2D Human Pose Estimators: Strengths and Caveats

Submitted 28-06-2022, in partial fulfillment of
the conditions for the award of the degree **MSc Artificial Intelligence.**

Hugo Brouwer
6636659

Supervised by
Metehan Doyran, MSc.
Dr. i.r. Ronald Poppe

Department of Information and Computing Sciences
Utrecht University

Abstract

We provide a way to get a deeper insight into what the strengths and caveats are of state-of-the-art models in 2D multi person pose estimation on a public benchmark dataset. This is done by adding labels to the validation set of the COCO Keypoint Detection Task 2017. The added labels correspond to challenges within this field, namely: occlusion (divided into occlusion by: self, other person and environment), truncation by the image border, image resolution and wrong annotations. The new annotations are publicly available for other researchers to get a better insight into how their models perform on the validation set at `https://github.com/AgntBrwr/2d-human-pose-estimators-strenghts-and-caveats`. The performance of several state-of-the-art models on the new annotations is also analyzed. All of these newly added labels substantially influenced the performance of the models and the models tended to perform differently when faced with the specific challenges. Furthermore, the wrong annotations are also used to discover a pattern to use to filter the wrongly annotated data from the train set. The state-of-the-art models are trained on the filtered train set and the original train set to investigate the impact of wrongly labelled instances on performance in this field. This did not lead to a performance increase for any of the models, but more research is required to get a deeper insight into this topic.

Contents

Chapter 1

Introduction

1.1 Motivation

2D human pose estimation, a.k.a. human keypoint detection is an actively studied field in the computer vision community. The aim of 2D human pose estimation is to correctly predict the location of human bodily keypoints (e.g. ear, knee, wrist, etc.) [29]. The ability to correctly predict these locations enables advancements in the development of multiple applications, such as activity recognition [7, 11, 24], human-computer-interaction [7, 29], robotics [24], animation [29] and behaviour understanding [18]. Moreover, 2D human pose estimation can be divided into two sub-categories: single person pose estimation and multi person pose estimation. Single person pose estimation focuses on predicting the keypoints of solely one person and multi person pose estimation focuses on the same but for multiple people. The latter is more difficult since it adds the task of correctly mapping each keypoint to the correct person [27]. In this study the focus will be on multi person pose estimation, since it is the more difficult task, and it includes and extends single person pose estimation. Another distinction is between 2D human pose estimation and 3D human pose estimation. Where the former tries to predict the keypoints in 2D-space, the latter tries to predict it in 3D-space. The addition of 1 dimension in 3D pose estimation is more difficult, since for the same 2D pose there are multiple 2D interpretations. Moreover, the field is rapidly moving towards 3D models. However, there are still challenges to overcome in 2D human pose estimation. Overcoming these challenges benefits the 3D pose estimation as well, since there are approaches in this field which infer 3D coordinates from 2D coordinates [5, 33]. So, this research will focus on the challenges to be overcome in 2D human pose estimation.

Furthermore, human pose estimation in general has rapidly advanced. Similarly as other subdomains of computer vision, the rapid advancements were possible due to the use of deep networks [30, 31] and accessibility of new manually labelled datasets [1, 17]. Still, despite these rapid advancements the state-of-the-art (SOTA) models face challenges, which lead to the formulation of new research problems within the pose estimation community. Current models struggle with keypoint estimation in images which include different types of occlusion, low image resolution, truncation at the image border and variation in the way humans and their keypoints appear [1, 2, 17, 27]. Several of these contemporary challenges for SOTA models are addressed by different benchmark datasets on which these models can be trained and compared with other models. These benchmark datasets include images containing specific challenging conditions for the models to generalize better and predict the keypoints with higher accuracy [17]. However, there is no method to get a clear insight into how SOTA models perform under these challenging conditions. A benchmark dataset providing detailed insight into particular challenges gives the opportunity for future research to focus on why and how general and specific strengths and caveats arise in the models. Besides, the future research can focus on how to overcome the caveats, which leads to the general aim of this study.

The general aim of this study is to contribute to filling up this knowledge gap and discover what the strengths and weaknesses are for the current SOTA models in the human 2D pose estimation field. This will be done by annotating the validation set of the COCO Keypoint Detection Task 2017 with the following labels: three types of occlusion (self, other person, and environment), truncation by the image border, image resolution and wrong annotations. A subsidiary aim is also to look at to what extent wrongly labelled data influences the performance of SOTA models in 2D human pose estimation. This can be investigated by filtering out wrongly labelled data from the train set and validation set. Furthermore, this dataset is chosen since many researchers already published their results on this dataset and it fits the best with our research aim. A more detailed description on this choice is given in Chapter 3. In short these specific labels are chosen, since these are re-occurring problems as problems for SOTA models in 2D human pose estimation according to the literature and these problems can be labelled in such a way to minimize ambiguity. A more detailed description of why these labels are chosen will be provided in Section 3.3. These labels will help to reveal certain strengths and caveats of current SOTA models. Moreover, an analysis of the new labels is given in Chapter 4. Then the results of the experiment will be presented and discussed in Chapter 5 and 6 respectively. Finally the pitfalls of this research will be discussed as well potential future studies in Chapter 7. Consequently, we hope that this study will high-light promising directions for future research to address these limitations.

1.2 Research Objectives

The main objective of the research is to get an insight into what the strengths and weaknesses are for the current SOTA models in 2D multi person pose estimation. Specifically, with regards to the labels which will be added, which are: three types of occlusion (i.e. occlusion by self, another person or the environment), truncation by the image border, image resolution and wrong annotations. This leads to the following formulation of the main research question:

> **Main research question:** How do state-of-the-art models in 2D multi person pose estimation perform on specific challenges present in the validation set of the COCO Keypoint Detection Task 2017?

To answer the main research question, it will be broken into a sub-research question to investigate how each of the specific challenges affects the performance of SOTA models and a sub-research question to investigate if the SOTA models are affected differently by these challenges:

- **Sub-research question 1:** How is the performance of state-of-the-art models affected by:
  - Different types of occlusion? Specifically by
    - self-occlusion?
    - other-person-occlusion?
    - occlusion by the environment?
  - Truncation at the image border?
  - Image resolution?
  - Wrong annotations?
- **Sub-research question 2:** Do the state-of-the-art models differ in performance on these challenges?

To answer sub-research question 1 the validation set of the COCO Keypoint Detection Task 2017 will be annotated with the given labels, and the performance of a selection of SOTA models will be measured with regards to these new labels. For sub-research question 2 the performance of a selection of models on these labels will be compared. By answering these two research questions we can determine how SOTA models in 2D multi person pose estimation perform on different types of occlusion, truncation at the image border, varying image resolutions and wrong annotations present in the validation set of the COCO Keypoint Detection Task 2017.

Additionally, the subsidiary objective is to look at how wrongly labelled data influences the performance of SOTA models in the field of 2D multi person pose estimation. This leads to the subsidiary research question:

> **Subsidiary research question:** Does removing the wrongly annotated data instances in the train phase lead to better performance for SOTA models on the COCO Keypoint Detection Task 2017?

To answer this research question the train and validation set will be filtered from wrongly annotated data instances. For this the models will be split in two groups.

One group will be trained on the filtered train set and the other group on the original train set. Then the results will be compared on the validation set. For this to work the train set has to be filtered. To go through the whole train set manually is time consuming, hence the train set will be filtered automatically. To automatically filter the data a pattern has to be found in the wrongly annotated data instances and this pattern has to be applied to the train set. Therefore, while discovering wrongly annotated data during the annotation phase of the validation set the features of mistakes made will be judged to see what pattern arises.

1.3 Thesis Contribution

2D multi person pose estimation is a widely studied field. However, to the best of our knowledge there has not been a study yet into the strengths and caveats of the SOTA models to this extent for this domain. This is a knowledge gap to be filled with extensive study. This is done by, mostly manually, adding contemporary challenge labels to a public benchmark dataset. Then the performance of one baseline model and three of the current SOTA models will be analyzed on the extended dataset. In this case that means that performance will be measured separately per specific challenge with the use of the Object Keypoint Similarity (OKS) on the validation set of the COCO Keypoint Detection Task 2017. Moreover, the train set will be automatically filtered. Then each model will be trained separately on the filtered train set and on the original train set. These will subsequently be compared on the filtered validation set. Besides, the labels will be made publicly available for the community to get an insight into how their model performs on the contemporary challenges and give a possibility to compare their strengths and weaknesses with those of other models. Additionally, the filtered train and validation set will be published to contribute to the 2D human keypoint detection community.

On a final note, to guide this study we will review the current literature on current public benchmark datasets in the field of 2D multi person pose estimation and the corresponding SOTA models. Additionally, automatic filtering and then the methodology will be discussed. It will include the choices for what to annotate, as well as the selection of SOTA models, the choice for which dataset to extend, and how to approach the data filtering method. Then the new annotations and the filtered data are analyzed. Followed by a presentation and discussion of the results. Finally we will present the limitations and future work. So, the contributions of this study to the 2D multi person pose estimation community can be summarized as follows:

- An analysis of the performance of current state-of-the-art models in 2D multi person pose estimation on the following challenges: three types of occlusion (self, person, and environment), truncation by the image border, different image resolutions and wrong annotations.

- Publicly available annotations on a public benchmark dataset for the community to get a better insight into the performance of their own model on these challenges.

- An insight into the influence of wrongly annotated instances in the train set on the performance of SOTA models on the COCO Keypoint Detection Task 2017.

Chapter 2

Literature Review

In this chapter we will present the literature with regards to 2D multi person pose estimation and data filtering. Hence, first we will review the literature on public benchmark datasets for 2D multi person pose estimation and their corresponding evaluation metrics. Then the SOTA models within this domain will be reviewed. At last, the literature on automatic filtering on data will be reviewed.

2.1 Public Benchmark Datasets for 2D Human Pose Estimation

The public benchmark datasets for 2D multi person pose estimation are Microsoft Common Objects in Context (MSCOCO, usually abbreviated as COCO) [17], Max Planck Institute for Informatics (MPII) [1], CrowdPose [16] and AI Challenger [34]. These will be discussed consecutively in this section. Note that each of these datasets tries to overcome some of the research problems within this research domain. So, for each of the different datasets the research problems they address will be described and per dataset a discussion will be provided on why it is a public benchmark. After that the metrics for the different datasets will be briefly discussed. An overview of the characteristics of the datasets can be found in Table 2.1.

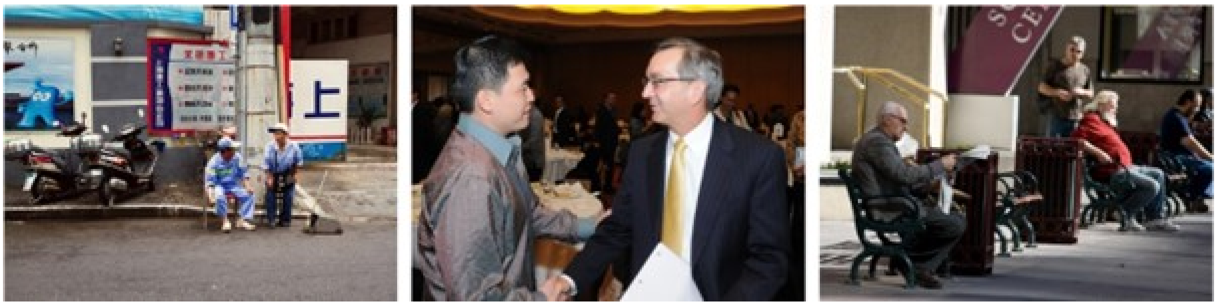


Figure 2.1: Example images from the COCO dataset with regards to the keypoint detection task. Pay attention to the presence of non-iconic views and requirement of contextual reasoning.

2.1.1 COCO Dataset

As for the discussion of the benchmarks for 2D human pose estimation, the datasets will be discussed in order. First, we will discuss the Microsoft Common Object in Context dataset. COCO is used as a public benchmark for several computer vision tasks, such as object detection, instance segmentation, image captioning and multi person keypoint detection [17]. It uses the same dataset for all the different tasks COCO. For all of these computer vision subdomains the emphasis of the COCO dataset lies on solving the task within the corresponding natural context. Such as predicting the segmentation of a car driving on the road, or predicting the keypoints of a person walking around a public square with other people in the background. The choice for solving tasks within context has been made, because SOTA models in various computer vision fields have difficulty when the corresponding task must be solved in context. The COCO dataset addresses 3 core research problems in image understanding for computer vision. These are detecting non-iconic views, contextual reasoning between objects and precise 2D localization of objects [17]. Here a non-iconic view means that the view does not contain a single large object in a canonical perspective (i.e. expected way to the object) centered in the view [17]. For computer vision models to be reliably deployed in more difficult computer vision tasks, the models should become better in detecting non-iconic views. Since these include contextual cues, views of everyday life, with potential partial occlusion or with clutter, which will also be very likely to be encountered once the models are deployed in the real world. Besides, contemporary SOTA models already achieve great performance on images with iconic-views [17]. Furthermore, contextual reasoning entails creating a mathematical model of a certain object and a relation with its context, which could lead to better reasoning about objects in context. And precise 2D localization is important to create a correct 2D representation of a 3D world.

For pose-estimation specifically, the full COCO dataset contains 250k annotated people. These images are gathered from Flickr of which example images can be observed in Figure 2.1. The COCO Keypoint Detection Task 2017 splits this data with 57k images for training, 5k for the validation set and 20k for the open test set. The 2017 task data split is the most relevant since it is the same split of data as the human keypoint detection challenges in the years that followed. Note that only the train and validation set annotations are publicly available. The test set annotations are private, to create fair competition between models. Furthermore, COCO uses 17 types of keypoints. The dataset also contains a wide variety of annotations, for example for occlusion. Where occlusion is per keypoint and each keypoint is given a flag. This flag can denote that a keypoint is visible, invisible, or unlabelled. COCO is one of the most used public benchmarks for 2D human pose estimation.

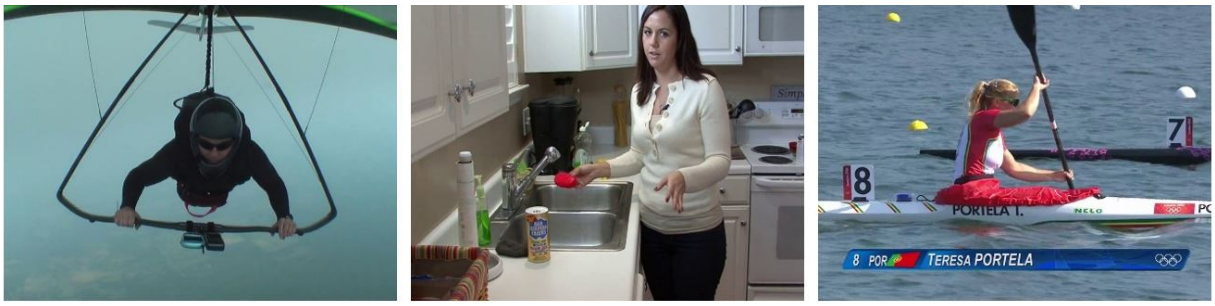


Figure 2.2: Three random example pictures from the MPII dataset. Take notice of the people in the picture being engaged in activities with different pose complexities.

2.1.2 MPII Dataset

The MPII dataset is smaller compared to COCO but is used as a public benchmark for multi person keypoint detection. The core research problems MPII addresses are appearance variability and complexity [1]. These are addressed by gathering a wide variety of human poses. The dataset contains an activity label for each person, covering 410 human activities This dataset had more labels described in the paper to analyze the performance of models under challenging circumstances, such as occlusion, varying part lengths, torso rotations and truncation [1]. However, to the best of our knowledge, besides pose complexity, these labels are not available to use anymore[1]. This makes it hard to get an insight into the performance of SOTA models under difficult circumstances.

This dataset, MPII, contains 40k annotated people in 25k images. The images are gathered from YouTube videos of which some example images can be observed in Figure 2.2. Of these images 18k are used for training and 7k for testing. Since the creators of the dataset also want to have a fair comparison of results the test set annotations are not publicly available. Note here that there is no official validation set for the MPII dataset. Furthermore, MPII contains 16 types of keypoints. Each of these keypoints has a visibility flag: either labelled or unlabelled. Other than COCO, to the best of our knowledge this dataset does not include occlusion labels anymore. MPII is, with COCO, the most used public benchmark dataset within this field.

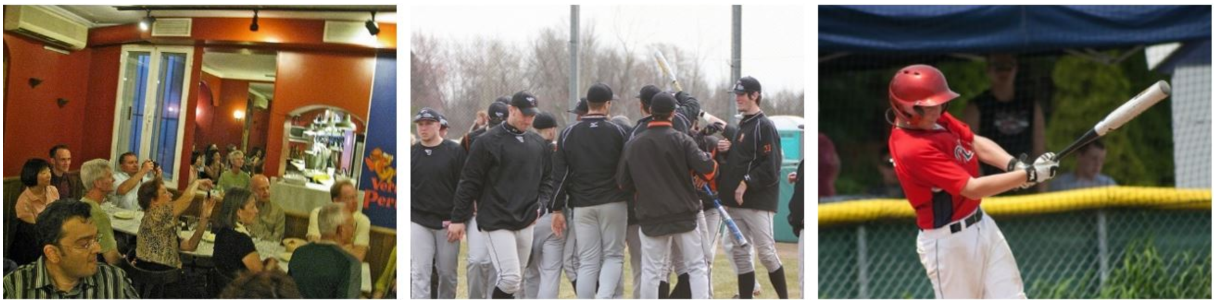


Figure 2.3: A representation of the pictures in the CrowdPose dataset in which a lot of pictures occur with overlap between people.

2.1.3 CrowdPose Dataset

Besides COCO and MPII we have CrowdPose. The CrowdPose is created specifically for multi person pose estimation. The research problem CrowdPose addresses is the problem of correctly estimating poses in crowded scenes [16]. The dataset is created to increase

[1] http://human-pose.mpi-inf.mpg.de/#download

performance of SOTA models in crowded scenes and advance the research in this area. Some example images of CrowdPose can be observed in Figure 2.3. Note that to solve crowded cases the most important observation is that a crowded scene is not necessarily an image with a lot of people in it, but rather an image in which there are a lot of overlaps between people. This is indicated in the CrowdPose paper by the Crowd Index. The idea behind including more crowded situations is that datasets like COCO and MPII contain a lot of situations representing daily life. So, the datasets include a lot of images consisting of only a few people without many overlaps. Hence, the SOTA models solely trained on COCO, MPII and AI challenger overfit on uncrowded scenes and this leads to worse performance on images with crowded scenes [16].

CrowdPose is, similarly as MPII, relatively small and contains 20k images with 80k people. The images in the dataset are gathered from COCO, MPII and AI Challenger. It uses 14 types of keypoints. It has the same labels as COCO but adds a Crowdindex to indicate the crowding level. It is derived by averaging the crowd ratio of all persons in an image. Where the crowd ratio is calculated for one person by dividing the number of joints of other people in the bounding box of this person divided by the number of joints of this person.

Figure 2.4: Example images from the AI Challenger dataset in which the researchers main focus was to make a bigger dataset for human pose estimation.

2.1.4 AI Challenger Dataset

Finally, there is the AI Challenger dataset. This dataset can be used for human pose estimation and Chinese image captioning [34]. What this dataset adds is that it is the first large scale Chinese image captioning dataset, and it is the largest 2D multi human pose estimation benchmark. It contains 300k images containing 700k people gathered from internet search engines [34]. Some example images can be observed in Figure 2.4. The core idea of this dataset is that due to the size of the dataset and the dependency of deep networks on convolutional neural networks (CNNs), it reduces the risk of overfitting [34]. In every image each person is labelled with a bounding box and 14 types of skeletal keypoints [34]. Similarly, to COCO and CrowdPose, this dataset has for each keypoint a visibility flag: visible, invisible, or unlabelled. However, even though it is the largest 2D multi human pose estimation dataset, the AI Challenger dataset has not been used much as a benchmark. This is partly due to a more practical issue, namely that the dataset is hard to access outside of China which makes it hard to use for a large group of researchers. Additionally, it does not introduce new challenging situations compared to earlier constructed datasets.

Dataset	Images	People	Keypoints	Source	Evaluation Metric
COCO	200k	250k	17	Flickr	OKS
MPII	25k	30k	16	YouTube	PCKh
CrowdPose	20k	80k	14	COCO, MPII, AIC	OKS
AIC	300k	700k	14	Internet search engines	OKS

Table 2.1: Overview of the number of images, number of annotated people, number of types of keypoints, source of images and the standard evaluation metric of COCO, MPII, CrowdPose and AI Challenger.

2.1.5 Evaluation Metrics

The evaluation metrics used to compare the results of the keypoint detection task differs for the benchmark datasets. COCO, CrowdPose and AI Challenger use the same evaluation metric, but MPII uses a different metric. The former three datasets use average precision (AP) and average recall (AR) as evaluation metrics. The AP is the percent of correctly predicted targets averaged over the classes. While the AR is the percent of true targets found by the model, averaged over the classes. In other words, the AP is the number of true positives divided by the sum of the true positives and false positives, averaged over the classes and the AR is the true positives divided by the sum of true positives and false negatives, also averaged over the classes. The AP and AR can both be calculated given the Object Keypoint Similarity (OKS) in a similar way as with the intersection over union (IoU) [23]. Where the intersection over union is given by dividing the area of overlap by the area of union. The OKS is the average keypoint similarity across all labelled items. Hence, the OKS functions as a threshold, similarly the IoU. Furthermore, here OKS is defined as:

$$OKS = \frac{\sum i[exp(-d_i^2/s^2k_i^2)\delta(v_i > 0)]}{\sum i[\delta(v_i > 0)]} \tag{2.1}$$

Here d_i is the Euclidean distance between the ground truth coordinates and the predicted coordinates. The d_i is passed through an unnormalized Guassian with standard deviation sk_i. Here s_i denotes the object scale and k_i denotes the per-keypoint constant which is a constant to control the influence of the difference between true and predicted location on the similarity score. The visibility labels are decoded as v_i and can have either value 0, 1 or 2 depending on whether the keypoint is unlabelled, invisible, or visible respectively. Hence, when a keypoint is unlabelled but predicted by the model, it does not affect the OKS score. When the OKS is near 1 it means that the predictions are near perfect and when the OKS is 0 the predictions are off by more than a few standard deviations sk_i. Given the OKS it is possible to compute AP and AR similarly as IoU allows[2]. In a nutshell this means a prediction is a True Positive, when three conditions are satisfied: the confidence score is higher than the confidence threshold, the predicted keypoint matches the ground truth keypoint and the OKS score is higher than the OKS threshold. When either of the latter two conditions is not met, it counts as a False Positive. And in the case a keypoint is labelled but not predicted it counts as a False Negative.

The most important evaluation metric used is AP with threshold $OKS = .5 : .05 : .95$. AP can also be calculated as a loose metric with $OKS = .5$ or with a strict metric $OKS = .75$. Humans have near perfect performance for $OKS = 0.5$ with 95%

[2] https://cocodataset.org/#keypoints-eval

[17]. From $OKS = 0.85$ and onward human annotators have significant disagreements about the correct keypoint coordinates [27] and human performance drops rapidly after $OKS = 0.95$ [17]. The AP can also be calculated for medium objects or large objects, i.e. $32^2 < area < 96^2$ and $96^2 < area$ respectively. Note that the calculation of the OKS is done with a normal configuration of k_i, i.e. $k_i = 2\sigma_i$. Where the standard deviations for each keypoint in the COCO dataset differ and are empirically discovered for every individual keypoint. Furthermore, the standard for the COCO Keypoint Detection Task 2017 is that it allows the models to have 20 top-scoring predictions per image. The predictions are then ordered by their confidence scores from high to low. These will then be respectively matched to the ground truths for which they have the highest OKS score with. And when all matches are found, the matches will be evaluated with respect to the specified OKS threshold. Similarly, the same metrics, with the same restrictions can be calculated for AR. Note that very person is weighted equally when computing the AP and AR, regardless of the number of labelled keypoints.

The standard evaluation metric used by MPII is the probability of correct keypoint (PCK) metric. The idea is to check whether the joint position matches the ground truth by using the fraction of the overlapping bounding boxes as a threshold [1]. Even more precisely, the standard evaluation metric MPII uses is the PCKh, which is a slight modification where the matching threshold is defined as 50% of the head segment length.

2.2 State-of-the-Art Models for 2D Human Pose Estimation

All the SOTA models use deep neural networks nowadays to train on public benchmark datasets, such as COCO, MPII, CrowdPose and AI Challenger for 2D multi person pose estimation. Deep neural networks are used, since models that use this technique can learn richer representations compared to models with hand-crafted representations [32]. In early deep neural network break-throughs for computer vision, based on LeNet-5 [14], the general strategy was to go from high-resolution to low-resolution and then use the network to classify. Later strategies, such as the Residual neural network (ResNet) model [9] added skip-connections to allow successful training of deeper neural networks for image feature extractors in general. The Hourglass model [21], went from high-resolution to low-resolution and then tried to recover the high-resolution from the low-resolution. Many of the current SOTA models are inspired by High-Resolution Net (HRNet) [10, 11, 20, 36]. This model maintains a high resolution throughout the whole processing phase. This is done by gradually adding high-to-low resolution streams and connecting the multi-resolution streams in parallel [29, 32]. HRNet is a single person pose estimator which can be combined with a person detector for the task of multi person pose estimation. A lot of methods achieve SOTA results by extending HRNet. This is done by improving data-pre-processing [10, 29], improving the heatmap encoding-decoding performance [36], or post-processing methods [20]. These methods are not tied to HRNet and can be used to run with other models as well. The models for multi person pose estimation can be roughly divided into two approaches: top-down methods and bottom-up methods. This section includes a description of both approaches and descriptions of some of the SOTA models for each approach. To conclude this section a description is given about heatmaps, because heatmaps are the most common method for coordinate representation in human pose estimation used by many of the SOTA models [36].

2.2.1 Top-down Methods

Top-down methods first use a person detector to find every person in an image. Then these methods use single person pose estimation for every detected person to predict the keypoints [7, 11]. A simple, yet effective top-down method for human keypoint estimation is to extend the ResNet model [9, 25, 35]. The ResNet model is a model used often for image feature extraction in general. This method will be, similarly as some of the other methods discussed shortly after, described in more detail later in this section. Besides ResNet, one of the most extended SOTA top-down methods, HRNet, maintains high-resolution representations throughout the whole process. This is done by adding high-to-low resolution streams throughout the process and connecting the streams within the process [29, 32].

By extending HRNet other methods have achieved current SOTA results. One of these methods is the Distribution-Aware Coordinate Representation of Keypoint (DARK, a.k.a. DarkPose) method. This method is model-agnostic, but combined with HRNet it was able to get SOTA results. The method corrects problems associated with coordinate decoding from, and encoding to heatmaps [36]. Another method that extends HRNet and achieves SOTA results is OmniPose, which uses multiple components to increase its performance. It uses multi-scale representations to incorporate contextual information. It uses the WASPv2 waterfall module to increase the field of view without decreasing the image resolution, which leads to the extraction of more contextual information. It also incorporates Gaussian heatmap modulation to achieve more accurate and robust locations compared with an interpolation method by smoothing the heatmap image to decrease the influence of single or smaller groups of pixels [2]. The DarkPose module combined with HRNet and OmniPose achieve the highest scores on the multi person pose estimation task of the COCO dataset [3].

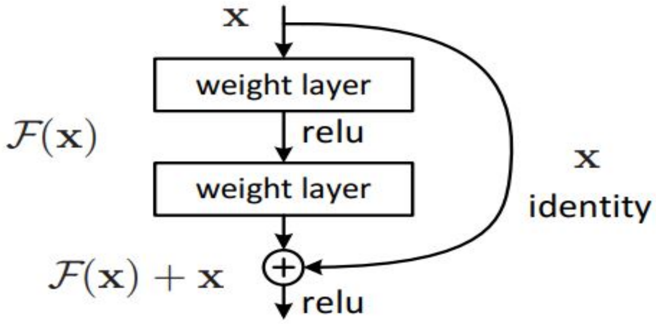

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2.5: Example of a skip-connection, a.k.a. a residual learning block in a ResNet model from [9]. These skip connections perform identity mapping and add the original input via the identity branch of some layers with the output of those layers via the normal convolution branch. This combats the problem of losing the effect of weights positioned early in the network [9].

ResNet

ResNet uses skip connections to optimize deeper networks more easily [9]. Before ResNet it was hard to do this, because deeper networks using backpropagation suffer from an

[3] https://paperswithcode.com/sota/multi-person-pose-estimation-on-coco

increasingly smaller gradient through hidden layers which is also known as the vanishing gradient problem [3]. Or simply, earlier layers in a network learn much slower, or even nothing, compared to later layers. The skip-connection, described in Figure 2.5, alleviates the vanishing gradient by using the identity function to prohibit the gradient from becoming exponentially smaller. One ResNet model specifically designed for human pose estimation is by [35], and adds some deconvolution layers at the last convolution stage of the ResNet model. In order to estimate the heatmaps from deep and low feature maps in a simple manner by upsampling the image. A simple representation of this ResNet architecture can be observed in Figure 2.6. This ResNet model is a top-down method so it first detects the people by using the faster-RCNN [26] and then it determines the keypoints for each person with the described architecture. The ResNet model designed for pose estimation has three versions: ResNet-50, ResNet-101 and ResNet-152. Here the 50, 101 and 152 stand for the number of layers the version has. These versions are tested with input sizes 256 × 192 and 384 × 288 and the best performing mode for this model is ResNet-152 with input size 384 × 288.

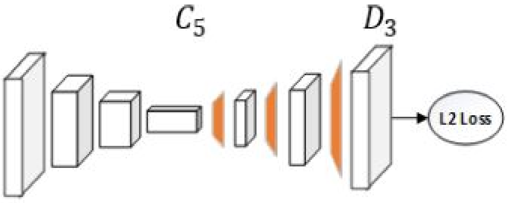


Figure 2.6: Simplification of the ResNet architecture from [35]

HRNet

HRNet maintains high resolutions by gradually adding high-to-low resolution subnetworks and connects these subnetworks in parallel [29]. A visual representation of a high-resolution network can be observed in Figure 2.7. The result of maintaining the high resolutions is that it is spatially more precise, since it does not have to recover coordinates from a downsampled image but is able to extract them from the high resolution image [32]. Furthermore, it is also a top-down method so it also uses a person detector. The person detector used here is the same as the previously discussed ResNet [35], which is the faster-RCNN [26]. From Figure 2.6 and Figure 2.7 can be observed that both models use upsampling, but ResNet does not maintain high resolution instances of the image, but solely re-creates this high resolution by upsampling. This model has two versions available: HRNet-W32 and HRNet-W48. Here the W stands for the widths of the high-resolution subnetworks in the last three stages [26]. That means that HRNet-W32 has widths of the last three subnetworks of 64, 128 and 256, while HRNet-W48 has the widths of 96, 192, 384. Both of these models are available with input size 256 × 192 and 384 × 288. The best performing mode is HRNet-W48 with input size 384 × 288.

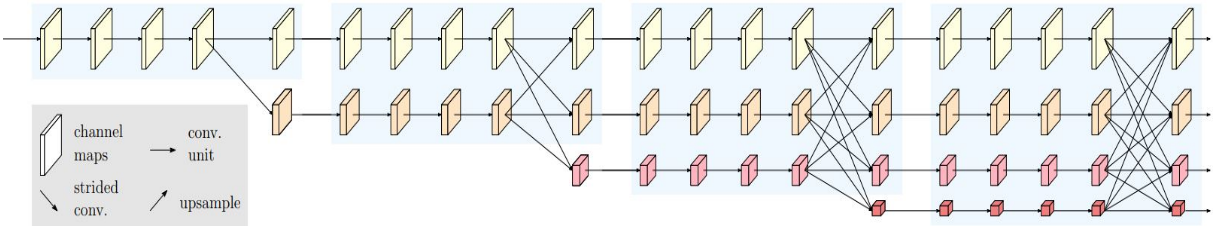


Figure 2.7: Example of a high-resolution network from [32]. Note that the network starts with a high-convolution stream and that it forms new stages every time it adds a new high-to-low resolution stream. Also note that the multi-stream resolutions are parallelly connected when a new stage is created. The connections from an upper layer to a lower layer are down sampled and from a lower layer to an upper layer are up sampled.

DarkPose

DarkPose is an model-agnostic method which aims to provide a better coordinate decoding- and encoding-method which does not rely on heuristic methods used prior to DarkPose [36]. So, it can be used for both top-down and bottom-up methods, but it is discussed here since it performed the best by extending HRNet. Furthermore, decoding means the translation of the predicted heatmap to the original image space, while encoding means the opposite, i.e. altering ground truth coordinates to heatmaps. For both decoding and encoding downsampling is required which leads to resolution reduction. For decoding downsampling is used, because images get downsampled to be computationally affordable, while for encoding downsampling is used to convert the image input size into the model input size. In turn, this downsampling can lead to the quantisation effect, i.e. mapping the values of one set to a smaller set. DarkPose tries to alleviate this for coordinate decoding in three steps which can also be observed in Figure 2.8. In the first step, heatmap distribution modulation, a Gaussian kernel with the same variation as the train data is applied to get a smoother heatmap, which can be used to efficiently calculate the first and second derivatives used in the second step. In the second step coordinates are calculated with Taylor-expansions. Finally, the last step is to recover the resolution from the heatmap to the original image space. For the encoding part the problem is alleviated by simply placing the heatmap center at the ground truth coordinate, instead of the calculated heatmap [36]. This is tested by extending the Hourglass model, ResNet and HRNet [36] with input sizes: 128 × 96, 256 × 192 and 384 × 288. It receives the best results by extending HRNet-W48 with input size 384 × 288.

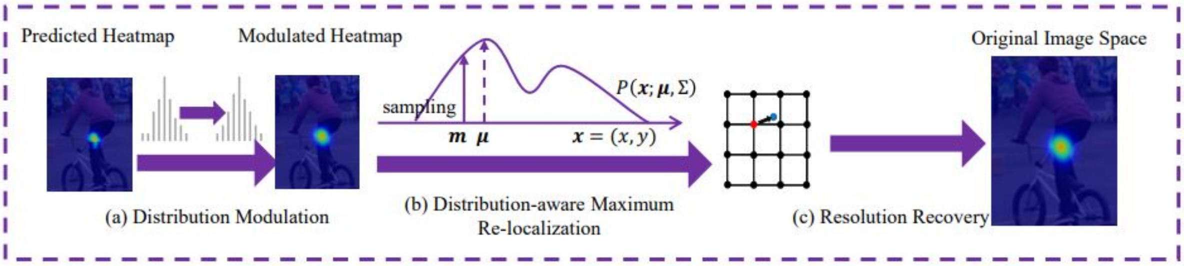


Figure 2.8: A visual representation of how DarkPose works from [36].

2.2.2 Bottom-up Methods

Bottom-up methods work by predicting individual keypoints and then grouping the keypoints to form individuals [7, 11]. Bottom-up methods are less accurate than top-down methods but are more efficient since models that use this method do not have the extra step of the person detector [7]. The same creators of HRNet also produced HigherHRNet, which is a bottom-up method for 2D multi person pose estimation and is based on the core component of HRNet, which is to maintain the high resolutions throughout the model [4]. Furthermore, HigherHRNet is extended further by the Disentangled Keypoint Regression

(DEKR) model [7]. This model builds on top of HigherHRNet by using branches where each branch represents one keypoint. Hence, every branch is specialized in learning and predicting a specific keypoint. This is supported by the adaptive convolutions which further enhance learning a specific keypoint by adapting the convolution size to the specific shape a keypoint can have [7]. A visualization of keypoint representations the model learns can be observed in Figure 2.9.

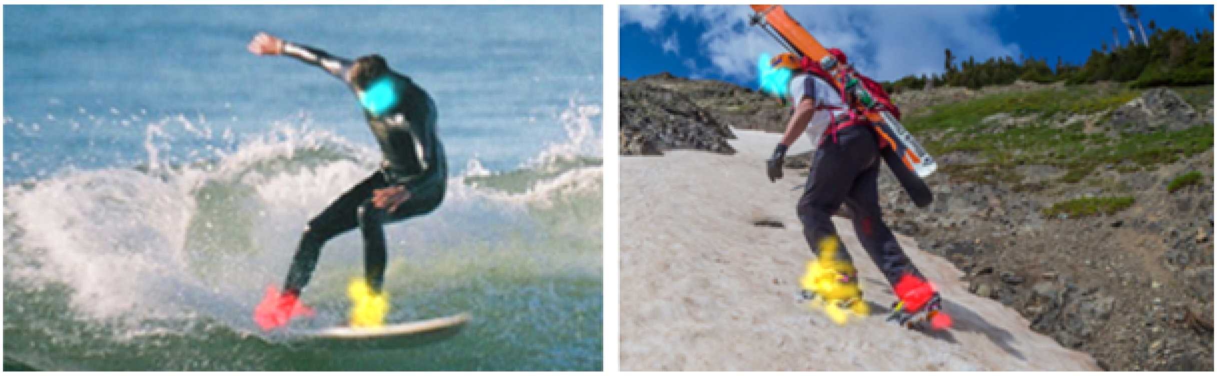


Figure 2.9: An example from [7]. A representation of how the separate branches will learn specific keypoints and that every keypoint has a common shape which supports the idea of using adaptive convolutions.

DEKR

DEKR is a follow-up of the HigherHRNet model and it consists of two parts: a multi-branch structure and adaptive convolutions. Due to the multi-branch structure it regresses separately for each keypoint: with every branch representing one keypoint [7]. And the adaptive convolutions make sure that the branch can adjust itself to learn about the specific keypoint region, as for example the keypoint region of a nose differs from that of a knee. Hence, the adaptive convolutions allow the model to learn to search for specific keypoint features in specific keypoints regions. Due to the combination of the multi-branch structure and the adaptive convolutions DEKR is able to regress for each keypoint individually and therefore it is able to achieve bottom-up SOTA results for correctly localizing keypoints. DEKR uses HRNet as its backbone. This means that HRNet extracts the feature map from the full input image, and then the DEKR method partitions the feature map for the keypoints to then use the multi-branch structure and adaptive convolutions to predict the keypoints. It uses input sizes 512×512 and 640×640. It gets the best results with HRNet-W48 as its backbone and an input size of 640×640. Furthermore, this model is tested with and without multi-scaling. Here multi-scale testing means heatmaps are produced at three scales: 0.5, 1 and 2, the regressed result is achieved by averaging over the heatmaps [7].

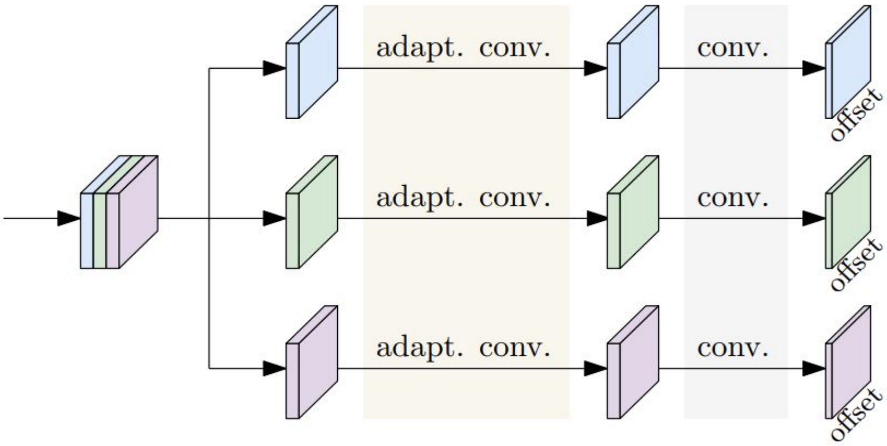


Figure 2.10: An example from [7]. A display of how the network is able to get the separate branches to learn specific keypoints.

2.2.3 Heatmaps

The most intuitive way to represent keypoints would be to just take the coordinates of the keypoints as a model output target. However, hardly any models use this approach. This could be due to the lack of spatial and contextual information in the representation [36]. An alternative is the heatmap. Heatmaps give spatial support around the ground truth location of the keypoints. This makes a model consider contextual clues and inherent target position. Hence, the potential reason that directly taking the model output target is solved, is because it may reduce the model overfitting during training [36]. Which is why heatmaps are the most standard way to use as coordinate representation when regressing keypoints during the train phase of the model [36]. A representation of how a heatmap represents the coordinates of a keypoint can be observed in Figure 2.11.

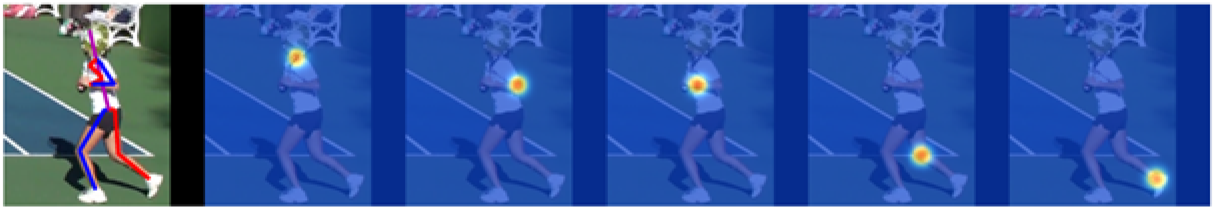


Figure 2.11: Example output from [21]. The most left image is the final output and then to the right are images with heatmaps of the neck, left elbow, left wrist, right knee, and right ankle respectively.

2.3 Automatic Filtering

A disadvantage of the (large) human-labelled datasets are the noisy labels [12, 22]. This can also be observed in the manually labelled COCO dataset where 1-2% of the data instances contain errors [4]. To train more reliable deep learning models it is possible to create better models, but it is also possible to purify the data from the noisy labels. In [22] a method was developed to automatically discover the error labels and filter these labels from the dataset. This has been applied in datasets such as MNIST [15], ImageNet [6] and Amazon Review [19] of which examples can be found in Figure 2.12. Moreover, the researchers found that filtering the noisy labels tends to lead to better performance, with simpler models increasing their performance the most. [8] found that with weakly-supervised deep learning, identifying the incorrectly labelled images and correcting them led to better performance for the deep learning models. However, the dataset here was relatively small (10.500 images) and using weakly-supervised learning there tend to be more mislabelled images due to automatic labelling compared to supervised learning where a dataset is labelled by hand.

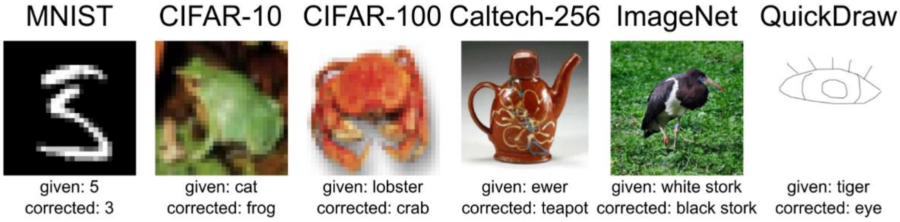


Figure 2.12: Different kinds of wrongly labelled data in different datasets from `https://github.com/cgnorthcutt/cleanlab`.

[4] https://cocodataset.org/#keypoints-eval

Still, even though automatic labelling leads to more noisy data, it is cheaper and faster than manually labelling the data by hand. That is why in [28] different methods to deal with noisy data were tested. To keep the noisy labels in the dataset has the advantage of resembling data gathered from the real world more in which the data is often noisy. Furthermore, the critique here on filtering the data is that it resembles real world data less, and also that too many true labels are filtered in the process. Which can negatively influence the performance of the model [28].

Chapter 3

Methodology

In this study we will run an experiment which will contribute to the research field of 2D multi person pose estimation. In this chapter we will discuss the components we will use in the experiment. First, we will describe the dataset. We will describe why the validation set of the COCO Keypoint Detection Task 2017 is chosen for the experiment and provide a small quantitative analysis of this dataset. Furthermore, we will discuss the algorithms. Here we will briefly explain why the baseline and the three SOTA models are chosen. Then the annotations are discussed. This is done by separately discussing the new annotations and explaining why these labels are chosen. To conclude this chapter the setup of the experiment will be summarized.

3.1 Dataset

3.1.1 Choice of Dataset

For our purposes we need to choose a dataset that many researchers already publish their results on and include sufficient challenges to label them. Hence, CrowdPose and AI Challenger do not seem to align well with the goals of this paper. These datasets have not been used as much in prior research compared to COCO and MPII. Besides, CrowdPose is labelled inconsistently (e.g. head and neck are always labelled as occluded as can be observed in Figure 3.1) and access to AI Challenger is difficult. As for COCO and MPII, both datasets have their advantages and disadvantages. For both, the best would be to annotate the test set to get the best insight into the performance of SOTA models on the contemporary challenges. However, test set annotations for both COCO and MPII are not publicly available. That is why we decided to annotate the validation set. Since MPII does not have an official validation set, the preferred choice is to annotate the COCO dataset. Besides, COCO is also larger compared to MPII. And most importantly, COCO also fits better with our research aim, since occlusion is an important contemporary challenge, and COCO already has visibility flags to get a basic insight into the performance of SOTA models with respect to occluded people. This is a potential starting point. The aim of this research is to get an insight into the strengths and caveats of human pose estimators by extending a public benchmark dataset. To reach this aim we chose to extend the COCO dataset, specifically the validation set of the COCO Keypoint Detection Task 2017. Future research could extend this research by annotating the test set of the corresponding dataset.

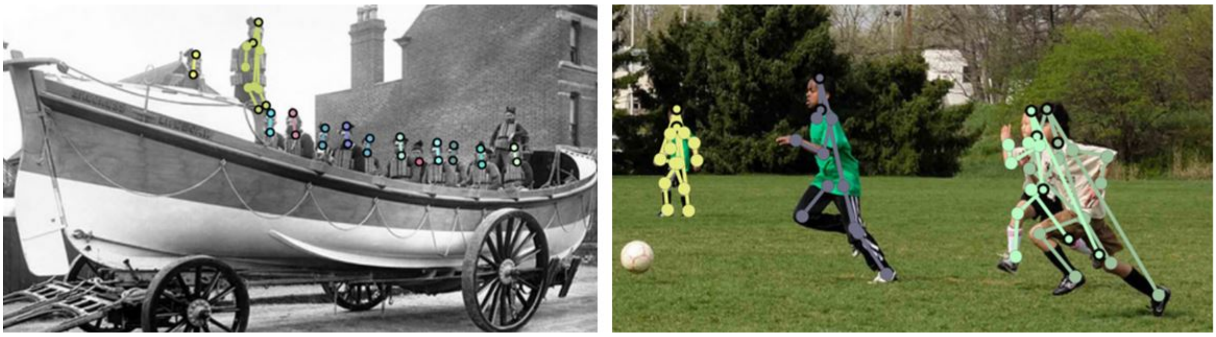


Figure 3.1: Examples from the CrowdPose dataset. In this dataset the head and neck are always labelled as occluded. This can be seen by the thick black edges around the keypoint ground truth locations.

3.1.2 COCO Validation Set

As the COCO validation set contains a subset of the COCO Keypoint Detection Task 2017, the validation set contains 17 keypoints as well. These keypoints can be seen in Figure 3.2) and represent the nose, left eye, right eye, left ear, right ear, left shoulder, right shoulder, left elbow, right elbow, left wrist, right wrist, left hip, right hip, left knee, right knee, left ankle and right ankle respectively. Besides the keypoint annotations every person has segmentation annotations, the visibility flag, image id, person id, category id, and bounding box. Where the image id is the specification of the image. Similarly, the person id is the specification of the person in the image. The category id is the specification of the object of interest, in the case of human pose estimation a person. And the bounding box is an imaginary rectangle in which the object of interest can be found.

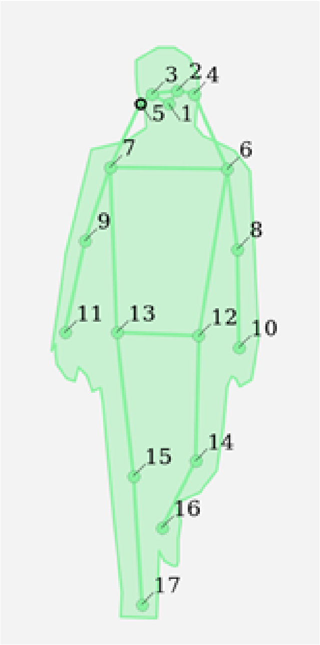


Figure 3.2: Visual example of the indexed keypoints in the COCO validation set, including the corresponding segmentation annotations. Here every keypoint is labelled and all, but keypoint 5 (right ear), are flagged as visible.

Furthermore, the COCO validation set for all computer vision tasks this dataset addresses contains 5.000 images. Of these 5.000 images, 2.693 contain annotated people. The 2.693 images contain 10.777 people with at least segmentation annotations. Of all these annotated people, 6.352 have been annotated with at least one keypoint. The distribution of people annotated with at least segmentation and annotated with at least one keypoint per image can be observed in Figure 3.3). In total 68.215 keypoints are annotated in the validation set. As previously mentioned, every keypoint is given a visibility flag; visible, invisible, or unlabelled. The distribution of the flags can be found in Figure 3.4). These labels can be used in comparison with the new occlusion labels and check the validity of the original labels.

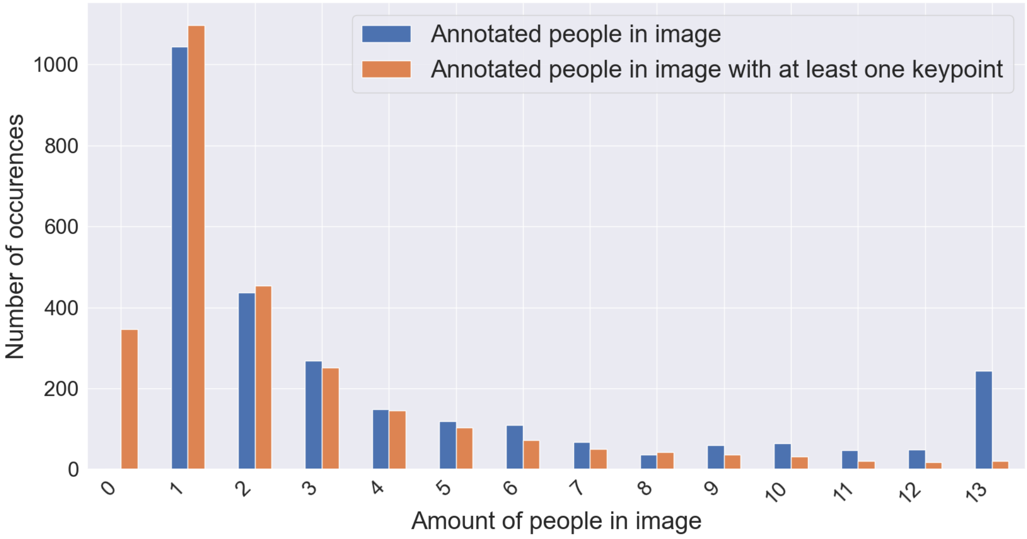


Figure 3.3: Distribution of the number of people in the images containing annotated people. It contains two distributions. The number of people in an image with at least the segmentation annotations and the number of people in an image with at least one keypoint annotation.

3.2 Human Pose Estimation Models

Four models are selected as benchmarks for the new labels. In general, top-down methods perform more accurately, but top-down methods potentially have different strengths and caveats with respect to certain challenging conditions. So, the inclusion of a bottom-up method might provide a better depiction of strengths and caveats with regards to specific challenges of SOTA models in general. However, due to complexity of the architectures of the models it is hard to fairly analyze and compare different models where approaches and models in which multiple components differ. We will also include a baseline model as one of the four models. As a baseline the ResNet model for human pose estimation [35] will be used, as it is designed to function as a simple baseline. Furthermore HRNet is included, because most SOTA results extend or use components of HRNet. The DarkPose module combined with HRNet is selected as the next model, since this combination is one of the best performers on the SOTA leaderboards for COCO multi person pose estimation. The last model is DEKR, a bottom-up method. It performs somewhat worse than SOTA top-down methods, but is one of the best performing bottom-up methods.

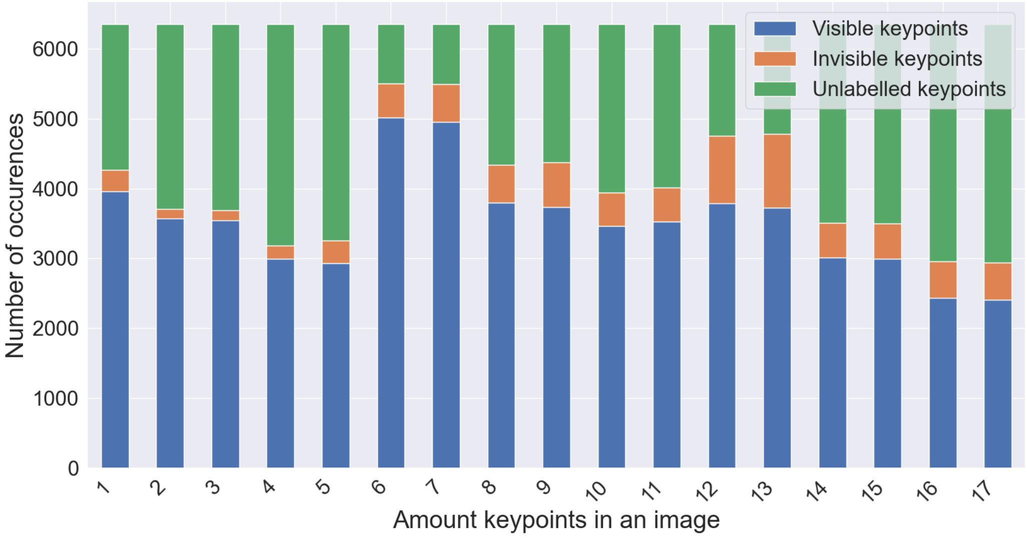


Figure 3.4: Distribution of the number of visibility flags. Each keypoint has one value. The value of the flag can either be: visible, invisible, or unlabelled.

3.2.1 Implementation details

Every model is trained and tested with a NVIDIA GeForce RTX 3060 12GB video card. The researchers of the trained and tested models already pre-trained them on the ImageNet classification task [6]. The original models are used for testing the models on the new labels, hence these models are only tested. For the filtered data part all models are trained and tested and the hyperparameters of these models only differ in batch size in comparison with the hyperparameters used in the original models. Instead of the original batch size of 32 for ResNet, HRNet and DarkPose, a batch size of 8 is used. And for DEKR 8 images are trained per GPU instead of 10. These choices are made due to hardware constraints.

Note that for the filtering method the models used in the experiments will also be trained and tested on the filtered data. The models trained on the filtered train set will be trained with the same hyperparameters as the models trained on the original dataset. An important thing to consider here is that the models trained on the original data have their hyperparameters tuned to maximize performance on these specific data instances. Hence, a performance increase with the filtered dataset with the same hyperparameters means that there is potential for an even better performance when these will be correctly adjusted. Moreover, the exact same hyperparameters cannot be used as the original models from the papers due to hardware constraints, the batch size is changed to still be able to train the models. The batch size is inherently tied to the number of epochs. An epoch E is when the machine learning model passes through all data once. And the batch size B is the number of samples the model goes through before the model is updated. And normally mini-batch mode is used, where $1 < B < N$, where N is the number of training instances, and one epoch is N/B iterations. Mini-batch mode is used here as well with a batch size of 32 for analysis of the new labels and 8 for the filtering analysis. If a batch size is too small it can lead to convergence at a local optimum. However, when the batch size is too big it can lead to bad generalization, but the reason why this is the case is unknown. Similarly, when the number of epochs is too low, the model will underfit, while too many epochs might lead to overfitting.

ResNet

The ResNet models of [35] are used. These consist of ResNet-50, ResNet-101 and ResNet-152 with input size 256 × 192 and the ResNet-152 also with 384 × 288. This model took 140 epochs to train with a batch size per GPU of 32. The Adam optimizer was used to optimize the model [13]. And the learning rate at 90 epochs drops from 1e-3 to 1e-4 and 1e-5 at 120 epochs. The model is downloaded from `https://github.com/leoxiaobin/deep-high-resolution-net.pytorchR`.

HRNet

Of the HRNet model several versions are tested: HRNet-W32, HRNet-W48 both with input sizes 256 × 192 and 384 × 288 from the [29] paper. Moreover, the training of the original models took 210 epochs for each model with a batch size of 32. To test the filtering method the number of epochs was kept the same, while the batch size was set to 8. To optimize the model the Adam optimizer was used, similarly as in the original versions. And it has a learning rate of 1e-3, 1e-4 and 1e-5 dropping at 170 and 200 epochs respectively. The model was downloaded from `https://github.com/`

leoxiaobin/deep-high-resolution-net.pytorch.

DarkPose

For DarkPose the versions DarkPose-W32, DarkPose-W48 are tested both with input sizes 128×96, 256×192 and 384×288. However, DarkPose-W48 with input size 256×192 was not trained, since it was not available. The rest of the hyperparameters are the same as HRNet, since DarkPose extends HRNet. Hence, 210 epochs with a batch size of 32, while for the filtering method a batch size of 8 was used. And, Adam was used to optimize the algorithm and the same learning rate as HRNet. The DarkPose model is downloaded from https://github.com/ilovepose/DarkPose.

DEKR

DEKR was tested with and without multi-scale testing for both input sizes 512×512 and 640×640. Furthermore, the Adam optimizer was used, and the model was trained with 140 epochs with a learning rate that decreased from 1e-3 to 1e-4 to 1e-5 at epochs 90 and 120 respectively. The only difference for the hyperparameters is that DEKR trains on 8 images per GPU instead of 10. The DEKR method was downloaded from https://github.com/HRNet/DEKR.

3.3 New Annotation Labels

There are several kinds of contemporary challenges for SOTA models in the 2D multi person pose estimation community. These challenges occur often in images from public benchmark datasets, but the tools to evaluate the performance of SOTA models on these challenges are scarcely available. We already discussed some of these challenges briefly: occlusion, lighting conditions, truncation at the image border, low image resolution, wrong annotations and variation in clothing. Other examples of challenges are interpersonal closeness [34] and cluttered images [17]. An important requirement to include a challenge as a label is that the challenge should be clear. For example, there is a clear way to annotate the occlusion labels, by just annotating the occluded keypoints by the specific type of occlusion. There is however also an option to label this vaguely, by ignoring the keypoints and mentioning how much occlusion is present by type in terms of ratio. This can lead to a higher error-rate in the labelling process, since the data will be manually annotated. This difficulty is even more problematic for challenges such as lighting conditions, where there is no clear, but only a vague way to annotate. Therefore, challenges such as lighting conditions, variation in clothing, interpersonal closeness and clutter will be left out. These challenges rely too much on subjectivity during the manual annotation process. This means that different annotators might label differently purely in certain situations based on their interpretation, especially during annotation of edge cases. Because in these cases the difference between two or more potential labels is small. Similarly, as pinpointing the exact moment a color goes from white to gray. Hence, the requirement for annotating is that the contemporary challenge can be annotated in a clear fashion, which minimizes subjectivity during annotation. This leaves the challenges which can be clearly annotated. So, the remaining challenges will be discussed. These are three types of occlusion (self, person, and environment), truncation at the image border and image resolution. Also, wrong annotations will be discussed, since these are present in every dataset, but are rarely mentioned in research. To attribute the labels to every person in the dataset with at least one labelled keypoint we will build an annotation tool. Finally, after the annotation process we will analyze the performance of the SOTA models on the new annotations with the COCO evaluation metric for human keypoint detection described in Section 2.1.5.

3.3.1 Occlusion

Occlusion, which is one of the most difficult challenges to overcome, is one of the annotations already included in the COCO dataset. In COCO the visibility flag denotes whether an object is visible, invisible, or unlabelled. Where invisible means occluded. The addition of this flag enables the possibility to see how well a model performs with occlusion. However, there is no insight into what kind of occlusion is present. Occlusion can occur by another person, by the person itself, or by the environment. Adding these different types of occlusion gives insight into how these different types influence SOTA models, if at all. Based on these insights one can decide to include new data or tweak a certain model to handle a certain type of occlusion better. The different types of occlusion present in the COCO dataset can be observed in Figure 3.5. In the case of self occlusion, the occlusion flag will be hardly given by COCO, as can be seen on the right image from Figure 3.5. Besides, the occlusion labels of the COCO Keypoint Detection Task 2017 are not of great quality. Hence, every labelled keypoint will be given new occlusion labels, and every occluded keypoint will be given one occlusion type label. This will lead to higher quality occlusion labels in general and extend the original dataset by including one occlusion type label for every occluded keypoint. Moreover, the occluded keypoints will be annotated solely as occluded when it is fully blocked. And if there are multiple types of occlusion applicable, only the type of occlusion is attributed which is closest to the camera. Hence, there will be minimal subjective influence in the manual annotation phase.

3.3.2 Truncation at the Image Border

Truncation at the image border also leads to lower prediction accuracy for SOTA models [1]. This is when an annotated person has keypoints outside of the image. In the left image of Figure 3.6 truncation at the image border can be seen. The idea is that accurately predicting keypoints gets more difficult the more keypoints are truncated by the image border. Here with the image border we mean the border of the full image without padding. This will be manually annotated per person by counting how many keypoints are truncated by the image border. Note that we will not count visible but unlabelled keypoints. When keypoints are visible in the image, but unlabelled When annotating for every person how many keypoints are truncated by the image border, the influence of subjectivity on the annotation process will be minimal. And keypoints are only labelled as truncated by the image border when these are fully truncated by the image border. This can lead to challenging situations, especially combined with other challenges, as the prediction from DEKR shows in Figure 3.7.

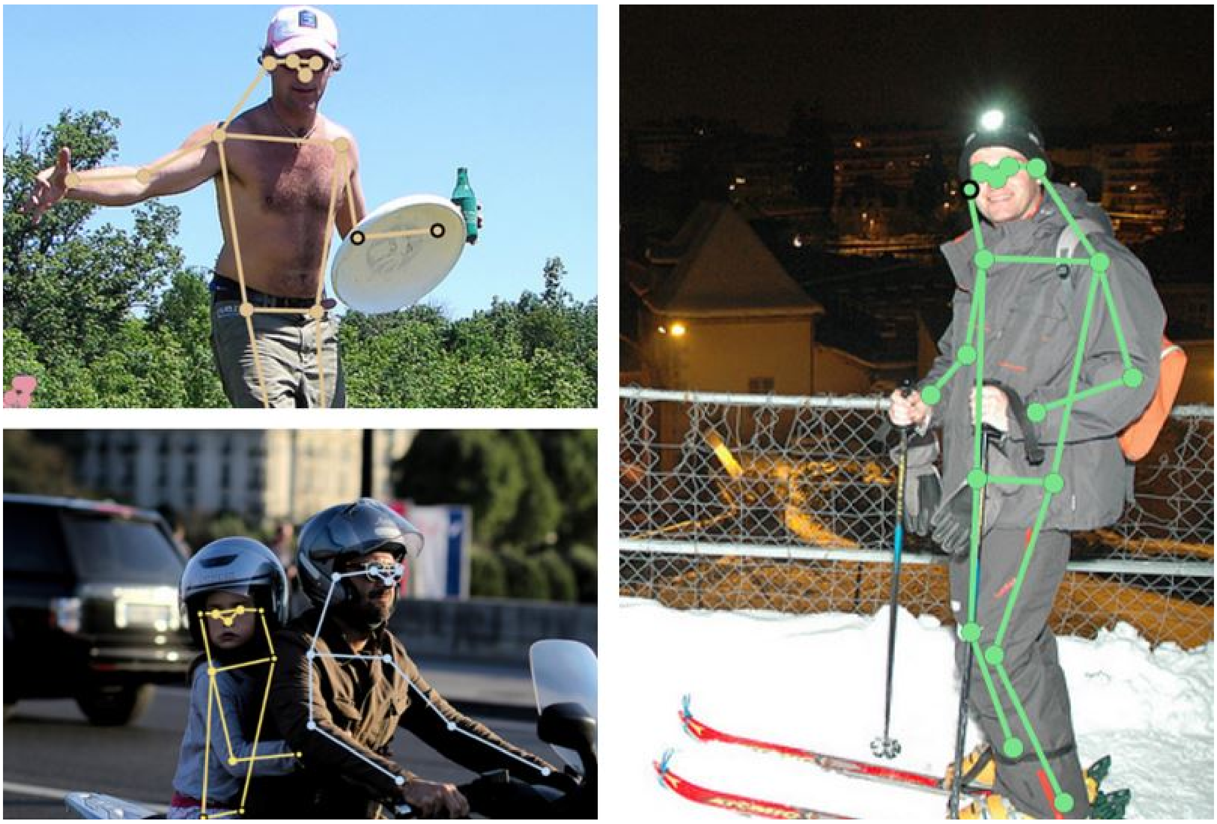


Figure 3.5: Different types of occlusion in images from the COCO validation set. **Top left:** In this image there is occlusion by the environment. The left wrist and left elbow are not visible as indicated by the COCO visibility flag. **Bottom left:** This image has other person occlusion in it. The left shoulder of the person indicated with the pink annotations is occluded by the person with the blue segmentation. The left shoulder is also flagged as invisible by COCO. **Right:** The image on the right contains self occlusion. The right knee and right ankle are for example occluded by the person. Note that this is not flagged by COCO.

3.3.3 Image Resolution

Image resolution can also be problematic. Current SOTA models maintain high image resolution throughout the process. The current SOTA models work better than previous models which maintain lower image resolution throughout the model and upscale it to a higher resolution. Which in turn also worked better than just lowering the image resolution. So, the prediction accuracy changes possibly due to degrading image resolution. The ablation study of [32] confirms that image resolution has an impact on the accuracy of keypoint prediction. Thus, to add an image resolution annotation to the data set gives an insight into the performance of different SOTA models when using different image resolution as input. COCO also includes lower resolution images, such as the right image

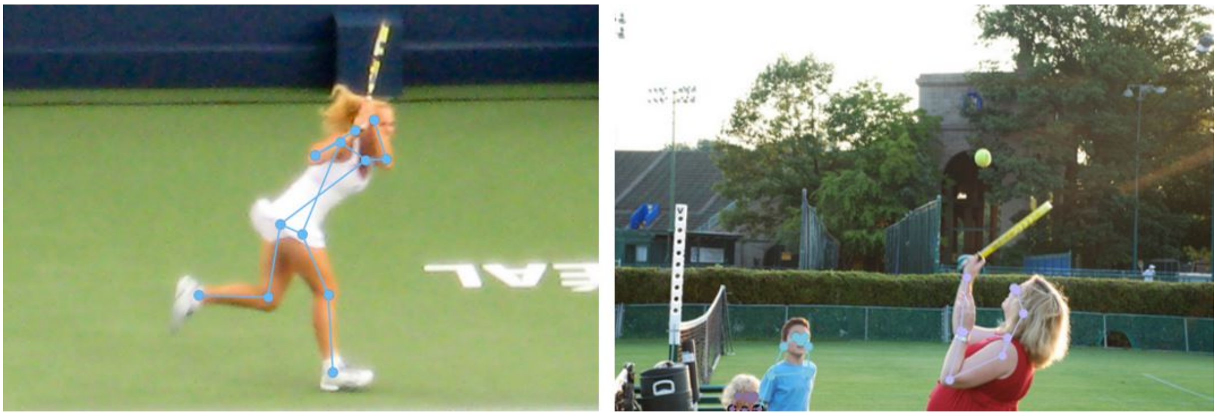


Figure 3.6: Example images from the COCO validation set. **Left:** This image has 3 people in it where all of them are truncated by the image border. **Right:** This image has a low resolution.

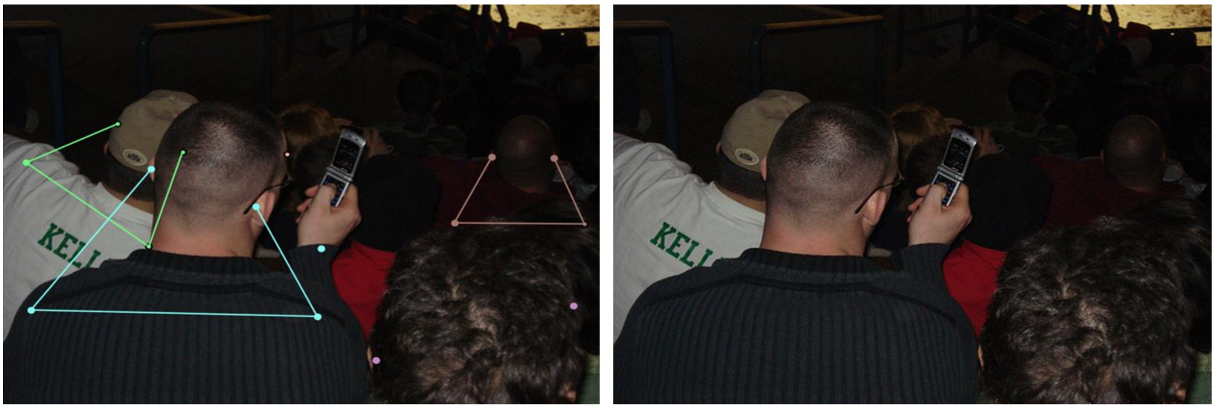


Figure 3.7: Example prediction from DEKR. **Left:** Ground truth keypoints. **Right:** DEKR did not predict any keypoints, while there are 5 people in the image with at least one keypoint labelled. However, this is a very challenging situation with both a lot of truncation at the image border and a lot of occlusion by other people.

in Figure 3.6. To minimize subjectivity different image resolutions will be chosen which can be calculated automatically. These resolutions will be labelled per person. Since a crop of an image in which a person is portrayed can be seen as an image with a resolution. The resolution per person is gathered by calculating the keypoints which stick out the most in a certain direction and then create the bounding box based on the minimum and maximum values in four directions (up, down, left and right). Two examples of this can be observed in Figure 3.8. Note here that since we do not use padding that not every pixel of the person falls within the bounding box, but the rest of the surface compensates for that. In the case that the distance between the upper and lower most, and right and left most keypoint is 0, the bounding box size in that direction is replaced by the square root of the person area size from the original dataset. This occurs in the case when either just one keypoint is annotated and can occur with multiple keypoints as well. The latter case is very unlikely, but when it occurs the keypoints on either the x- or y-axis are replaced by the square root of the person area size, depending on which axis has a difference of 0. However, there are multiple difficulties here. When the number of keypoints labelled decreases the more the bounding box size decreases which is not desired in this situation. Moreover, depending on the pose of the person the bounding box can also increase or decrease in size. These are disadvantages compared to the person area size used in the original dataset. Still, an advantage of this method is that constructing the bounding box like this is less prone to error than human drawn segmentations for persons.

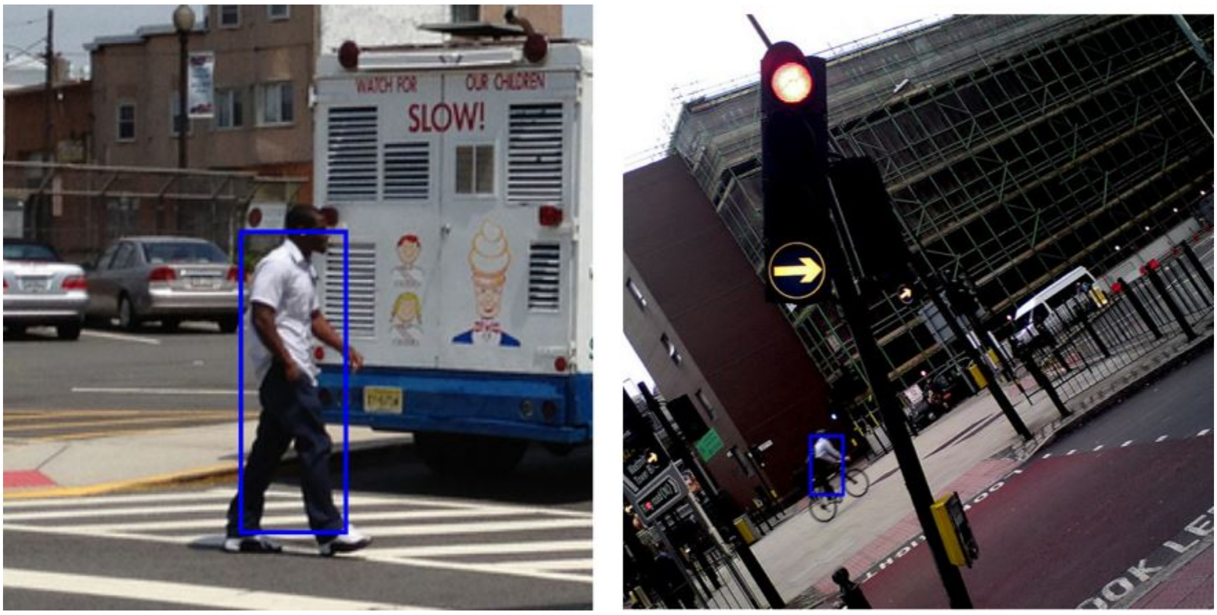


Figure 3.8: Calculation of the bounding boxes by calculating the minimum and maximum coordinates of the keypoints of one person on the x- and y-axis. The left image has a higher image resolution value here, since the images are of similar size but the bounding box is bigger than the bounding box in the right image.

3.3.4 Wrong Annotations

Another annotation to include is less straightforward and is hardly mentioned in research in the pose estimation community. This annotation is not a challenging situation, but rather something which makes accurately predicting the keypoints harder. It is annotating for every person whether this person has been correctly annotated or not. And the annotation process of keypoints is even more prone to human error than other subdomains of computer vision [17]. There are various ways in which labels are wrongly annotated in COCO specifically, as can be seen in Figure 3.9. The importance of including a label like this is two-fold. On the one hand, when a model performs worse on these labels than other models, the model might truly be better than can be observed from the misleading results. Since it might correctly label the keypoints, but due to a comparison with a wrong ground truth during the evaluation, it might seem to perform worse. On the other hand, the wrong annotations might lead to bias in training of the model. So, if a pattern can be found in the wrongly annotated data, a solution can be found to filter out the flawed data during training and create a less biased model. And this resulting model might perform better. This will be done by manually annotating for every person whether the keypoints are correct. Also, for every keypoint is checked whether it has the correct label. If not, they will be labelled as wrongly annotated. And for keypoints such as wrists, where there is a right and a left instance, these will be corrected if these labels are mirrored. Furthermore, to minimize the subjective aspect during annotation, small localization errors will not be labelled as wrong. Where a small localization error is not outside of the true keypoint of the human body and cannot be mistaken with another keypoint. Also, for images such as in Figure 3.9, the segmented people with missing annotations will be labelled as wrongly annotated instances. Furthermore, the wrong annotation label is also given in the case a keypoint is given the wrong keypoint label or when the location is too far off. The visible, but not annotated, keypoints in the image are ignored.

3.4 Filtering Method

A filtering method will be developed to remove wrongly labelled data from the train set. The filtered data will then be used to train the SOTA models and see whether the performance of these models will increase. To develop the filtering method we will inspect the wrongly annotated data from the validation set and try to discover a general pattern in this data. When a pattern is found, it will be used to create a method to automatically

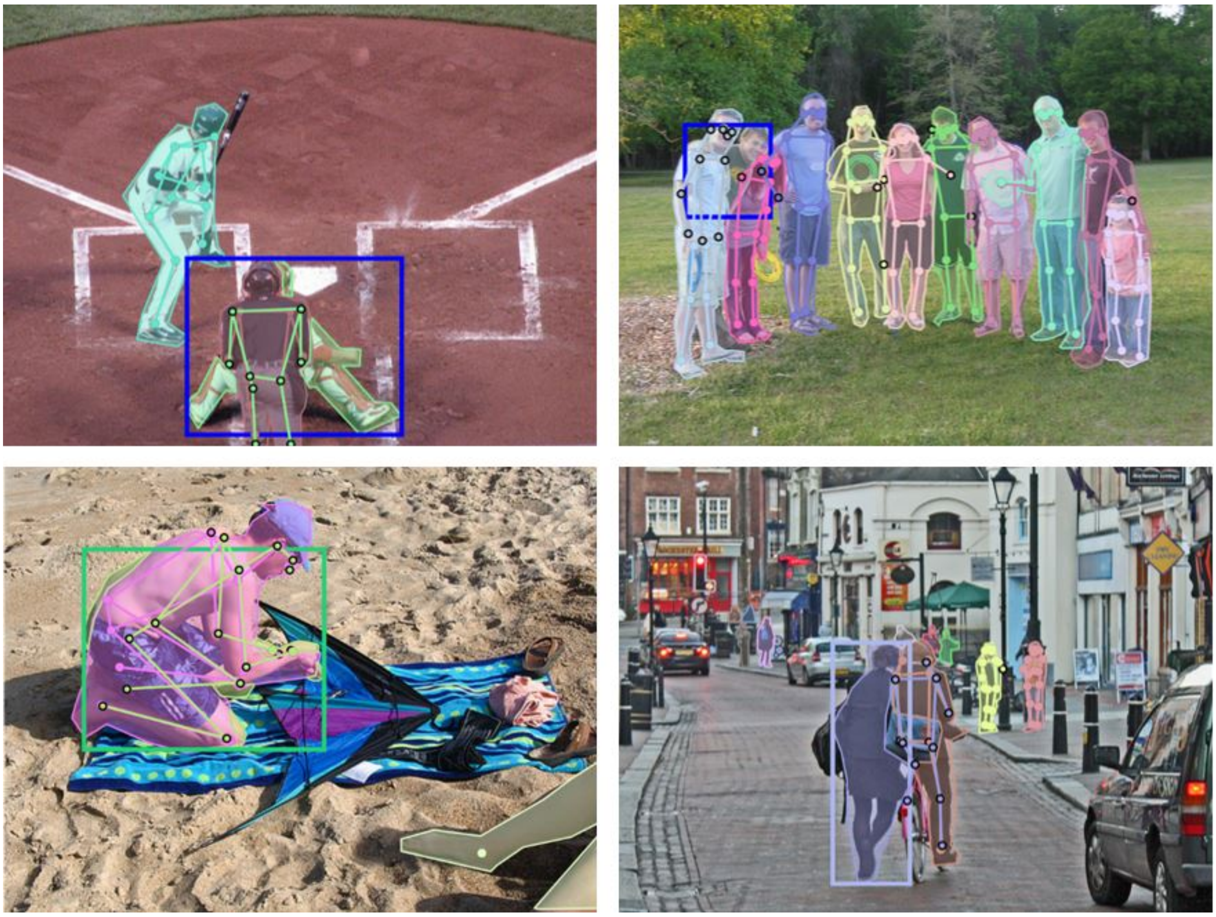


Figure 3.9: Instances of wrong annotations in images from the COCO validation set. The wrongly annotated people are surrounded by a bounding box. **Top left:** The person below with the green segmentation has been labelled as if it was the person with the pink segmentation. **Top right:** The person with the gray segmentation has the keypoints located on the person to his left. **Bottom left:** The person is annotated twice without another person missing annotations. **Bottom right:** The missing annotations of the person jumping on the back of the bicycle are located on the person riding the bicycle.

filter the wrongly annotated data from the train set. A disadvantage of filtering the data automatically is that it also filters true labels in the process. The idea here is to develop multiple automatic filtering methods and find one which can filter out at least 95% of the true positive wrong annotations in which the mistake is made which occurs most often. At the same time the method which accomplishes this, but also minimizes the number of false positives in the filtered data will be selected. Since the main goal is to filter out the 95% and the secondary goal is minimizing the false positives, the expectation is that there will still be a lot of false positives. Hence, we will go through the filtered data manually as well to separate the true positives and false positives. The advantage of this method is that the filtering method will not have a lot of true labels filtered in the end product. However, this method is based on a specific mistake made in the dataset, therefore it is dataset specific. The mistake which is often made is that the segmentation of a person does not correspond to the keypoints which can also be observed in the images at the top and the bottom right of Figure 3.9. This can be problematic during training, since the ground truth segmentation and bounding box are used as an indicator for the person to learn the keypoints for. And if these do not correspond to the keypoints, a bias can be introduced. This bias might lead to worse performance for the model. This will be investigated by analyzing the difference between the SOTA models trained on the filtered and the original train set.

Chapter 4

Data Analysis

To get a better insight into the annotation process and the new validation and train set these will be discussed in this chapter. So we start off with a description of the annotation process. Then the new labels will be analyzed per annotation type and other difficulties present in the dataset which are not annotated will be discussed. After that the wrongly annotated data from the validation set will be analyzed, and finally the data instances filtered from the train set by the filtering method will be discussed shortly.

4.1 Annotation Process

In the annotation process every person with at least one keypoint annotated in every image will go through the annotation pipeline. The first step in the pipeline is to manually determine whether a person is correctly labelled. In the case of a wrong annotation the person and image id are saved in the annotations to ignore during the evaluation. In the other case the process is split into the manual and automatic phase. During the manual process the decision is made for every labelled keypoint of a person whether it is occluded. For the keypoints labelled as occluded, a decision is made whether the keypoint is occluded by the person itself, another person, or any environmental factor. After that the number of truncated keypoints by the image border is counted. For the automatic part of the pipeline, the image id was stored, and after that also the person id. Then the image resolution was calculated per person, within the created bounding box. Finally,

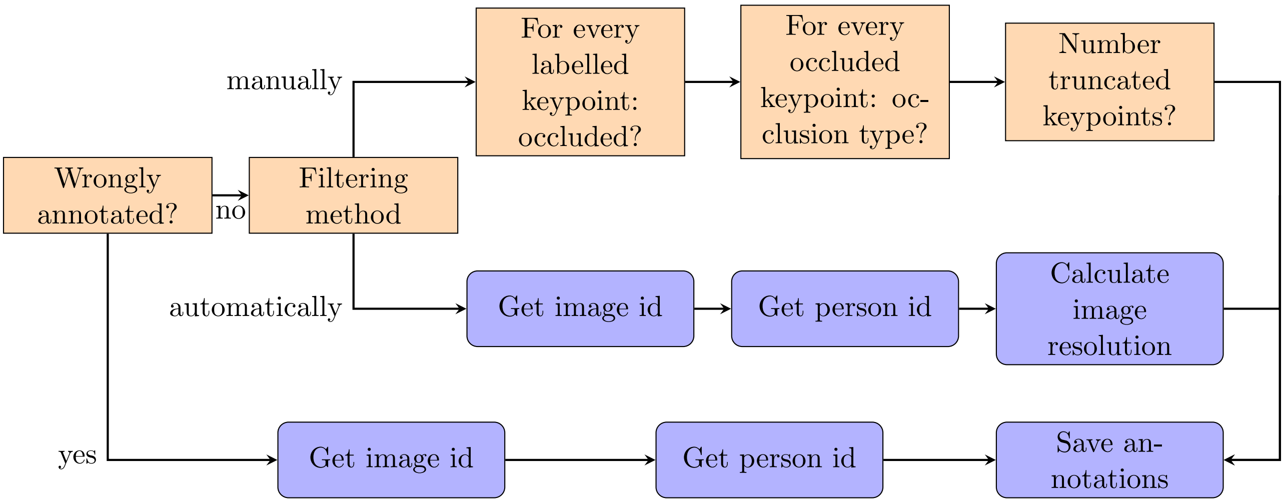

Wrongly annotated?
no
Filtering method
manually
For every labelled keypoint: occluded?
For every occluded keypoint: occlusion type?
Number truncated keypoints?
automatically
Get image id
Get person id
Calculate image resolution
yes
Get image id
Get person id
Save annotations

Figure 4.1: Flowchart of the annotation process. Here the orange boxes with sharp edges are manual processes and the purple boxes with round edges are automatic processes.

all data is gathered and saved, similarly as the COCO keypoint 2017 dataset, in a JSON object. The full process can also be observed in the flowchart in Figure 4.1.

4.2 Data Analysis New Labels

The newly added labels; occlusion, occlusion types, truncation at the image border, image resolution and wrongly annotated data will be analyzed separately in this section. The new annotations exclude 107 wrongly labelled people which leaves 6.245 people in the validation set. The general distribution of these new annotations can be observed in Figure 4.2 and Figure 4.3. The latter figure also contains the distributions in the case the keypoints are assigned one of the four keypoints groups: head, core body, arms or legs. The head keypoints consist of the nose, eyes and ears. The core body keypoints include the shoulders and hips. The arms contain the elbows and wrists. And the legs contain the knees and ankles.

4.2.1 Occlusion

The original validation set of the COCO Keypoint Detection Task 2017 already contained occlusion labels, but these occlusion labels were not labelled consistently and the labels were not of high quality as can be observed in Figure 4.4. Hence, the new occlusion labels

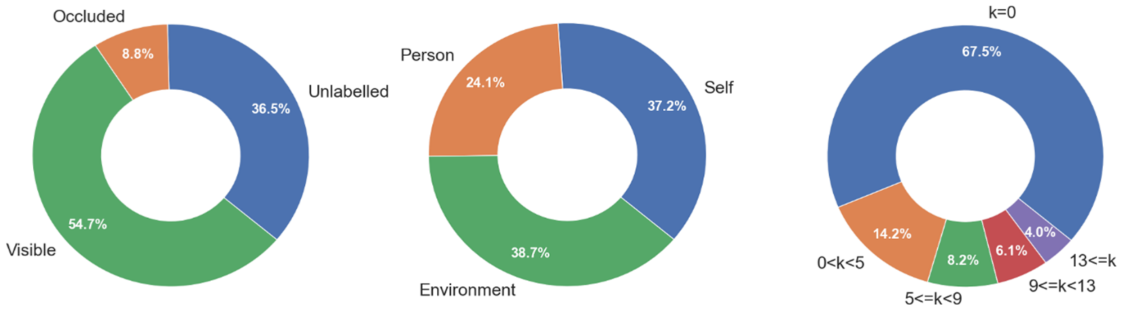


Figure 4.2: The distributions of the **Left:** new visibility labels: unlabelled, occluded and visible keypoints. **Middle:** type of occlusion: self occlusion, person occlusion and environment occlusion. **Right:** the keypoints truncated by the image border, grouped into no truncation ($k = 0$, where k is the number of truncated keypoints), between 0 and 5 occluded keypoints ($0 < k < 5$), between 4 and 9 occluded keypoints ($5 <= k < 9$), between 8 and 13 occluded keypoints ($9 <= k < 13$) and more than 12 truncated keypoints ($13 <= k$).

.

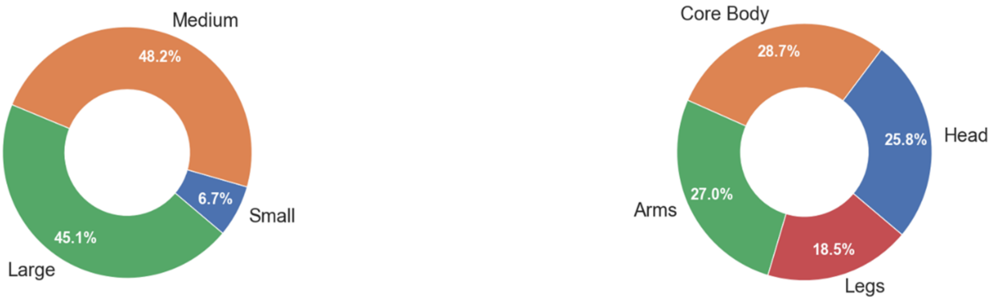


Figure 4.3: Distributions of the other labels: **Left:** image resolution of the person with bounding box resolution r: $r < 32^2$, $32^2 <= r < 96^2$, $96^2 <= r$ titled small, medium and large person. **Right:** the different groupings of keypoints: head, core body, arms and legs.

.

are compared to the original occlusion labels. Of the 6.245 people in the new annotations 38.747 keypoints are unlabelled, 9.364 are labelled as occluded and 58.054 are labelled as visible. In the original validation set only 11.54% of the labelled keypoints were labelled as occluded, but it is 19.46% for the newly annotated dataset. This difference is partly due to the inconsistency and quality, but also due to the inclusion of self occlusion, which was not labelled in the original validation set. Of the labelled keypoints 93.16% match between the original and new annotations of visible versus occlusion labels, in the case of excluding the wrongly annotated instances. And for solely occlusion labels 84.94% matches. The difference between the new validation set and the original for every keypoint as visible or occluded can be observed in Figure 4.5.

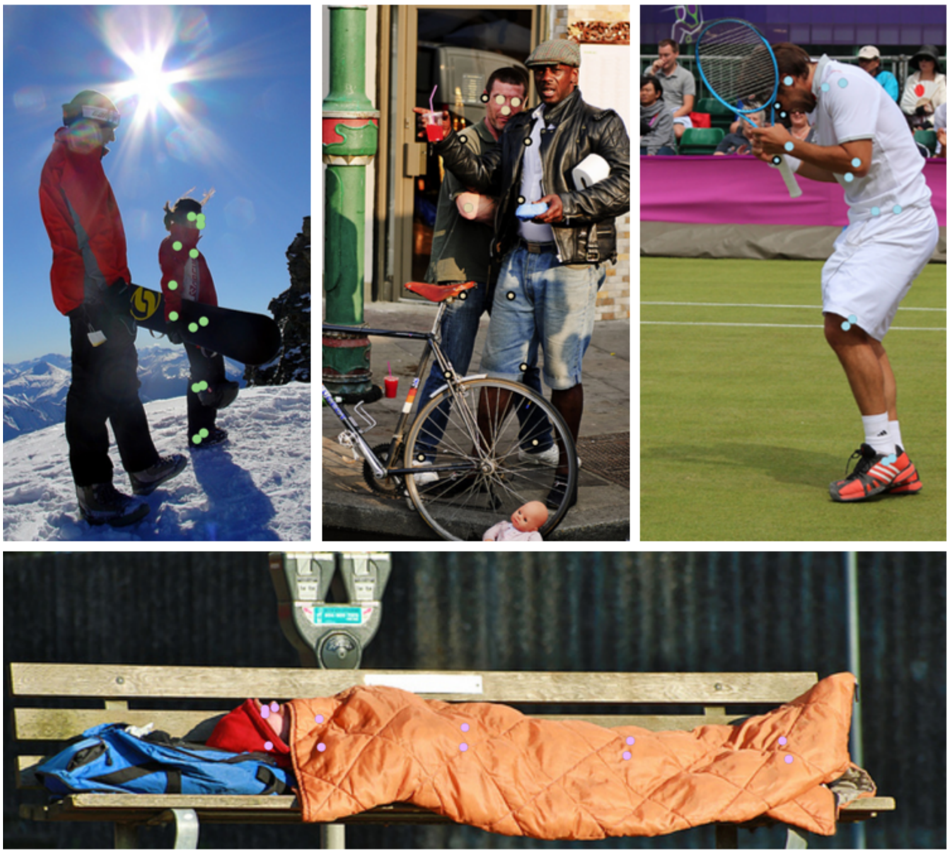


Figure 4.4: Different examples of wrong occlusion labels given to keypoints. **Top Left:** The snowboard occludes several keypoints, but none are labelled as occluded. **Top Middle:** The majority of keypoints are labelled as occluded, while most of them are visible. **Top Right:** The eyes, nose and right wrist are all labelled as occluded while these are visible, but other labels such as the right elbow are corrected as occluded since self occlusion is present. **Bottom:** Keypoints occluded by a blanket can be thought of as occluded, but also as not occluded. However, most keypoints under blankets were labelled as occluded, hence for the sake of consistency every keypoint under a blanket is corrected as occluded.

Summing the difference between the occurrences from Figure 4.5 shows that 1.583 more keypoints are labelled as occluded in the new occlusion annotations. Furthermore from the graph it can also be inferred that the biggest difference between the new and original annotations is that the shoulders and hips are labelled as occluded more often, while the nose is labelled as visible more often.

4.2.2 Occlusion Types

The occlusion type labels are newly added and consist of self occlusion, other person occlusion and environment occlusion. Of the 9.364 occluded keypoints 3.480 are self

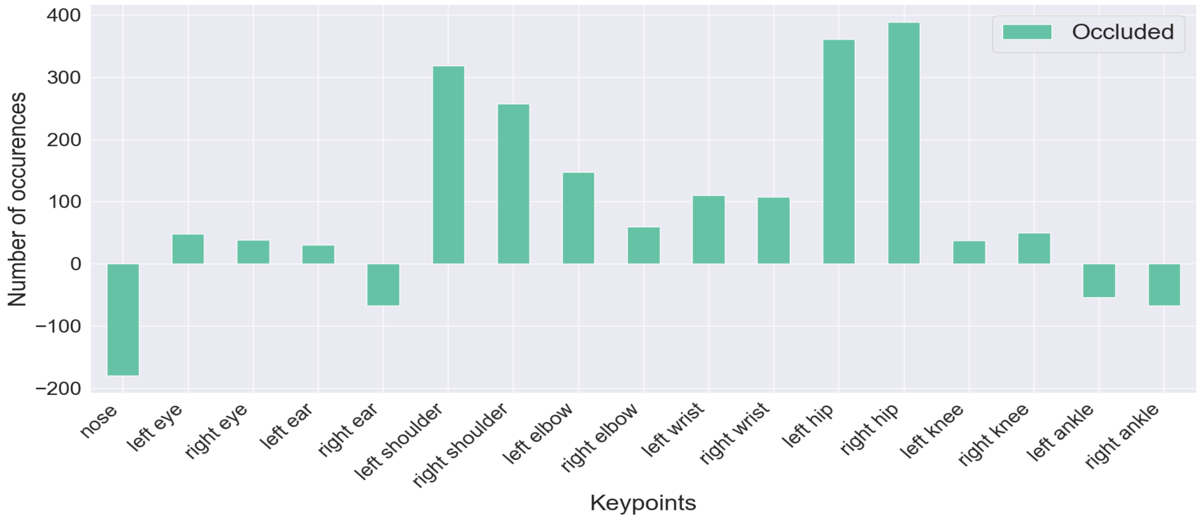


Figure 4.5: The difference between the original COCO validation set and the new validation set with regards to whether the labelled keypoint is occluded or visible. Here the positive number of occurrences is related to the number of occlusion labels occurring more often in the new occlusion labels for the keypoint at hand compared to the original labels. And vice versa for the negative number of occurrences.

occlusion, 2.263 are other person occlusion and 3.621 are environment occlusion. The distribution of the different occlusion types per keypoints is shown in Figure 4.6. What is remarkable here is that for the shoulders and arms self occlusion occurs relatively often, while for the head, hips and legs environment occlusion occurs relatively often. Especially for the shoulders, self occlusion is present relatively frequently. Other person occlusion occurs relatively the same to the number of occlusion labels per keypoint.

4.2.3 Truncation at the Image Border

Of the 6.245 people 4.217 do not have any truncation at the image border. The rest has at least one keypoint truncated by the image border and at most 16. The people get grouped together in bins of 4 based on the number of keypoints truncated by the image border to get a better overview of the results. This means that the people with 1 to 4 keypoints truncated by the image border are grouped together, etcetera. So, of the 2.020 people where at least one keypoint is truncated, the first bin contains 888 people, the second bin 506, the third 382 and the last 252. For each bin a heatmap is made based on whether a pixel in the image falls inside the corresponding person segmentation of the

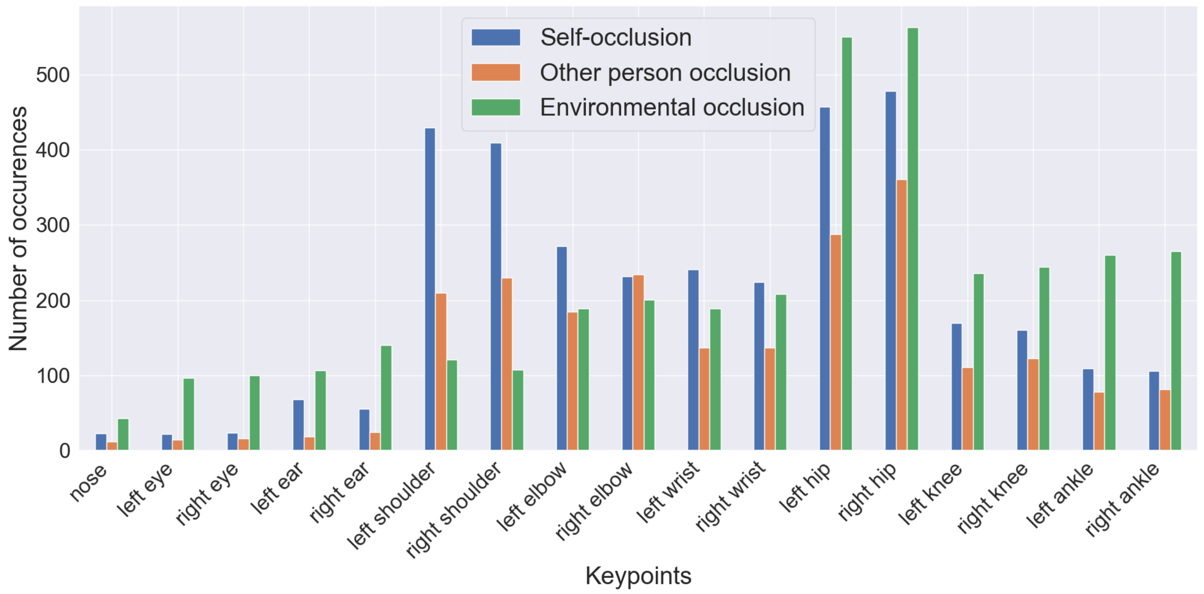
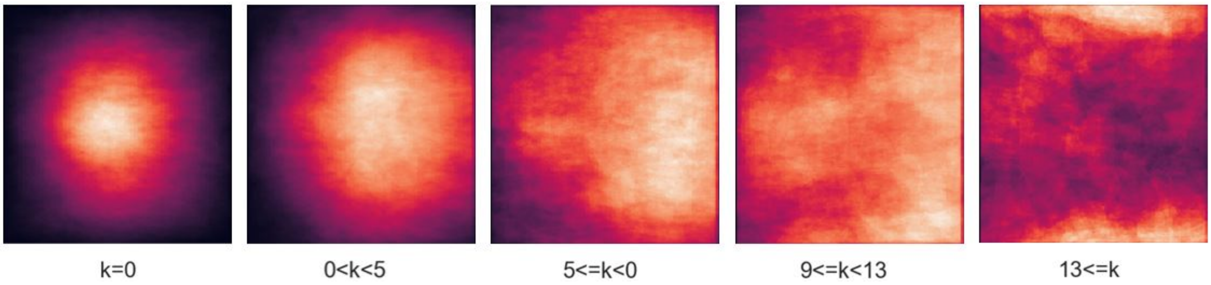


Figure 4.6: The number of occurrences per occluded keypoint with a split made for the different types of occlusion which can occur.

original validation set. These heatmaps are shown in Figure 4.7. Note that the heatmaps follow the intuition that people with no truncated keypoints, or a small number, are located more often in the middle and when the number of truncated keypoints increases the more people can be found at the edge of the image. Furthermore, the mean number of keypoints being truncated by the image border is 2 in the case for a person with at least one annotated keypoint. In the case where at least one keypoint is truncated by the image border the mean is 7.



k=0

0<k<5

5<=k<0

9<=k<13

13<=k

Figure 4.7: The likelihood of a pixel belonging to a person at a certain location in the normalized image split by the truncation bins, where k is the number of truncated keypoints for a person. In the lighter areas pixels corresponding to a person can be found more often in the corresponding location in the image for that specific bin.

4.2.4 Image Resolution

The image resolution for the bounding box of the person was divided into three bins with bounding box resolution r measured in pixels: $r < 32^2$, $32^2 <= r < 96^2$ and $96^2 <= r$. With occurrences of 421, 3013 and 2811 respectively.

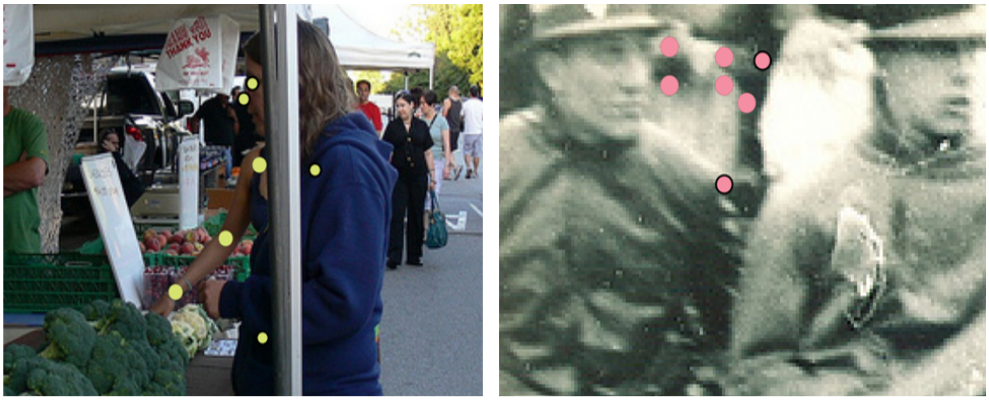


Figure 4.8: These two images include mistakes observed in the validation set. **Left:** Some keypoints are attributed to another person. The eye and nose are indeed occluded, but are exactly mapped to another person standing in front. **Right:** It is possible to see that there is a person sitting there, however it is a situation too ambiguous to judge where the keypoint are located exactly.

4.2.5 Wrongly Annotated Instances

Of the 6.352 people 107 have been wrongly labelled, i.e. 1.68% of the validation set has been wrongly labelled. Of these 107 instances, 76.64% has been wrongly labelled such that the person segmentation and bounding box do not correspond to the person keypoints. The person segmentation is determined by the bounding box and the bounding box is used as a person ground truth to learn the keypoints of one person. Instances of these mistakes can be observed in Figure 3.9. Other mistakes include: wrongly mapped keypoints within one person, attributing keypoints of one person to another person and mapping of keypoints in ambiguous situations, of which the first of these mistakes can be observed in Figure 4.8 and the latter two in Figure 4.9.

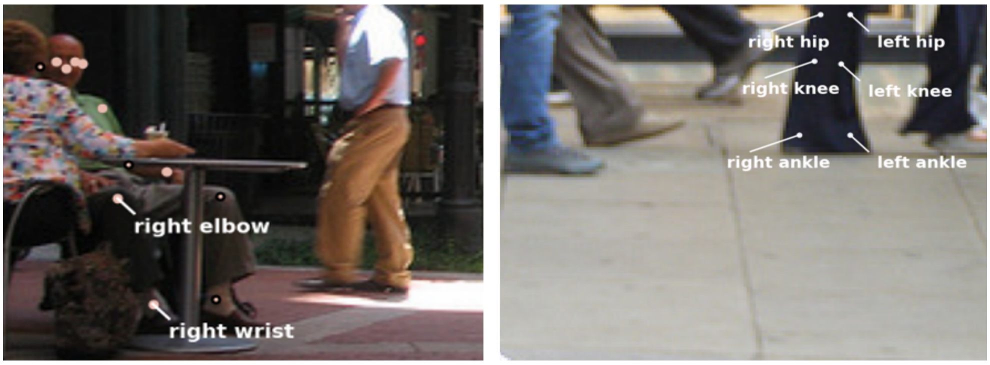


Figure 4.9: These two images include wrong mappings of keypoints within one person. **Left:** The elbow is labelled at a place where it is highly unlikely that the true location is there and the right wrist is labelled where the true location of the right ankle is. **Right:** The right leg of the person contains six labels. It seems ambiguous when solely looking at the leg itself, but from the context can be inferred that it is a right leg.

4.2.6 Other Difficulties

There are also other difficult situations occurring in the COCO Keypoint Detection Task 2017. These difficulties are not labelled, since these situations only occur rarely, but these lead to worse results. Examples of other difficult situations can be observed in Figure 4.10 and consist of difficulties such as distorted views of people, blur and other (for COCO) rare difficulties. Moreover, for some of these problems, such as the image with blur in Figure 4.10, it is not only difficult for an algorithm to correctly predict the keypoint, but it is difficult for people too. Which decreases human performance on a benchmark like this, but can also lead to distorted results due to errors in the ground truth annotation. A possible solution for this without filtering these data instances is to add another variable of certainty to the OKS formula. In which the certainty variable can increase the range in which a certain keypoint prediction can fall to get the same OKS score in an uncertain situation.

4.3 Analysis Filtered Data

To filter the wrongly labelled data from the train set the idea was to find a pattern in the wrongly labelled data from the validation set and this pattern has been found. In the validation set 76.46% is wrongly labelled with the same mistake discussed in the previous

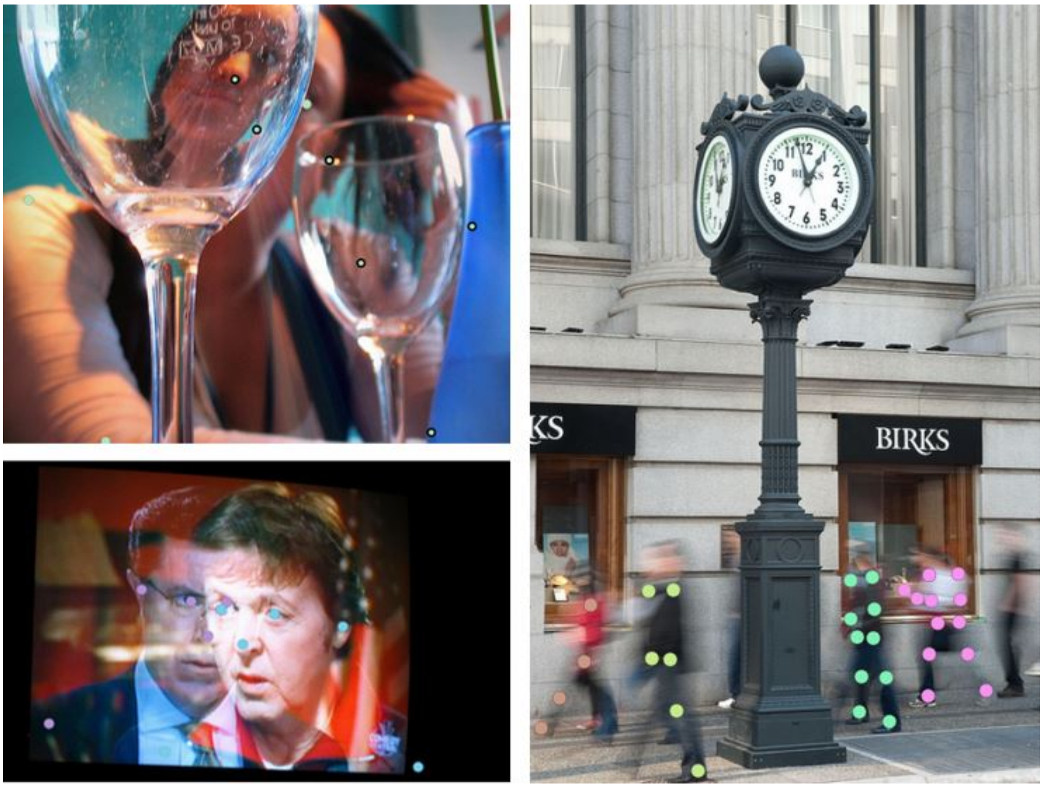


Figure 4.10: Multiple instances of other difficult circumstances for the models to correctly predict the labels which are not labelled during this annotations process. And these are quite rare. **Top left**: distortion of the shapes of a person through a transparent object. **Bottom left**: two people are video-edited in a way they are mixed up. **Right**: multiple people are blurred in the picture.

subsection. Since this mistake is made because of how the data is annotated for this dataset specifically, it cannot be used for other datasets besides CrowdPose which uses the same annotation API. The main goal of the filtering method was to develop a method to filter out 95% of the images in which the specific mistake was made. The secondary goal was to minimize filtering of true labels. To achieve this several methods were tested with different thresholds and some models achieved to filter out more than 95% of the wrongly annotated data, and of these, the one that minimized the number of false positive wrong labels was selected. This method was designed to take two characteristics into account. The first was whether a certain threshold was reached of keypoints labelled as occluded relative to the number of keypoints labelled as visible. Since, with this type of mistake the keypoints are often outside of the corresponding bounding box, which leads to the keypoints being annotated as occluded in the original dataset. The selected model did this by verifying whether at most 6 keypoints were labelled as visible. Then it verified whether the labelled keypoints consisted of at least 55% of occluded keypoints, or 50% when the person had more than 5 keypoints labelled as occluded. The second characteristic was whether a threshold was reached of keypoints falling inside the segmentation area of one of the other people in the picture. The selected model did this by verifying whether at least 90% of the keypoints of one person fell into the segmentation area of another person. With both of these thresholds several were tested until the main and secondary goal were reached. The selected method filtered 96.34% of the wrongly annotated data in the validation set, but even though this was the best model of the filtered data only 36.65% were actually true positives. This supports the idea that automatic filtering methods also prune too many true labels [28]. Hence, to alleviate this problem, i.e. decrease the number of false positives, these instances were manually labelled as wrongly annotated or not. In the case a person was wrongly annotated, the person was deleted from the annotation file. So, 4.042 of the 149.813 people with at least one annotated keypoint were automatically filtered in the train set and after manually pruning this group of people 874 were left to be deleted from the train set.

Chapter 5

Experimental Results

This chapter is dedicated to the presentation of the results. The results that will be presented can be divided into two parts. Part one contains the performance of ResNet, HRNet, DarkPose and DEKR on the new labels and part two consists of the results of the newly trained models with and without the filtered train data. The metrics used here are 6 of the 10 metrics discussed in Section 2.1.5, namely the AP and AR with various OKS thresholds (i.e. $0.5 : .05 : .95$, $.5$ and $.75$). In line with the guidelines[1], the primary metric is AP at $OKS = 0.5 : .05 : .95$.

5.1 Results for New Annotations

The results are gathered for the modes of the models discussed in Section 3.2.1. In this section only the best modes of the models will be discussed. Hence, that means for ResNet the ResNet-152, for HRNet the HRNet-W48 and for DarkPose the Darkpose-W48. These three models all use input size 384×288. Finally the DEKR model uses the DEKR-W48 with multi-scale testing enabled and input size 640×640. Example predictions of these models are shown in Figure 5.1. The full results of all variants of the models can be found in Appendix 8. The aim here is to see how the SOTA models perform on the newly labelled specific challenges in the COCO dataset. So, we will start with presenting the general results and then continue with the results per new label. In short, that means

[1] https://cocodataset.org/#keypoints-eval

Method	Data	AP	AP^{50}	AP^{75}	AR	AR^{50}	AR^{75}
ResNet-152	Filtered	0.752	0.905	0.821	0.805	0.945	0.868
	Original	0.743	0.896	0.811	0.797	0.937	0.858
HRNet-W48	Filtered	0.771	0.914	0.837	0.820	0.949	0.879
	Original	0.763	0.908	0.829	0.812	0.942	0.871
DEKR-W48	Filtered	0.732	0.892	0.796	0.786	0.934	0.843
	Original	0.723	0.882	0.786	0.776	0.924	0.832
DarkPose-W48	Filtered	0.777	0.914	0.843	0.825	0.949	0.884
	Original	0.769	0.907	0.833	0.817	0.942	0.875

Table 5.1: Performance comparison of the best modes of the tested methods on the filtered and original validation set of the COCO Keypoint Detection Task 2017.

that we will discuss the general results followed by the results on: occlusion, occlusion type, truncation, image resolution, wrong annotations respectively.

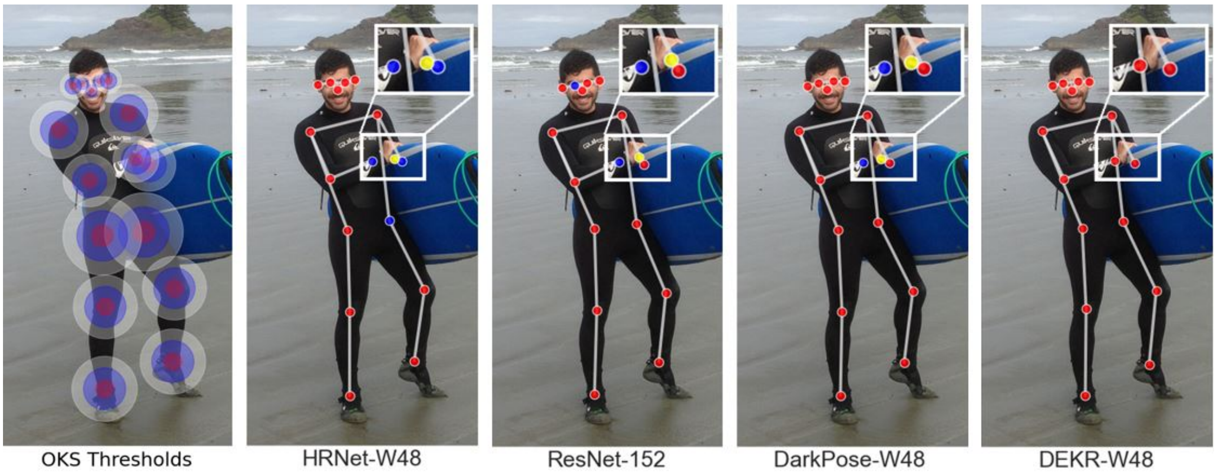


Figure 5.1: The left image contains the areas around the ground truth labels in which the models can obtain an OKS score of 0.5, 0.75 and 0.95 for white, blue and red respectively. Then to the right are the predictions of the models. In the images with predicted keypoints the colors correspond to the scores mapped to the OKS scores. And the yellow predicted keypoints do not affect the OKS score, since the unlabelled but predicted keypoints do not influence the score.

5.1.1 General

The general results on the filtered validation set tend to follow the same pattern as the original results and the performance is better for every evaluation metric as can be observed by comparing the results on the new and original data in Table 5.1. Furthermore this table also demonstrates that DarkPose performs the best, closely followed by HRNet and then by ResNet and DEKR respectively. This is the same order as the original

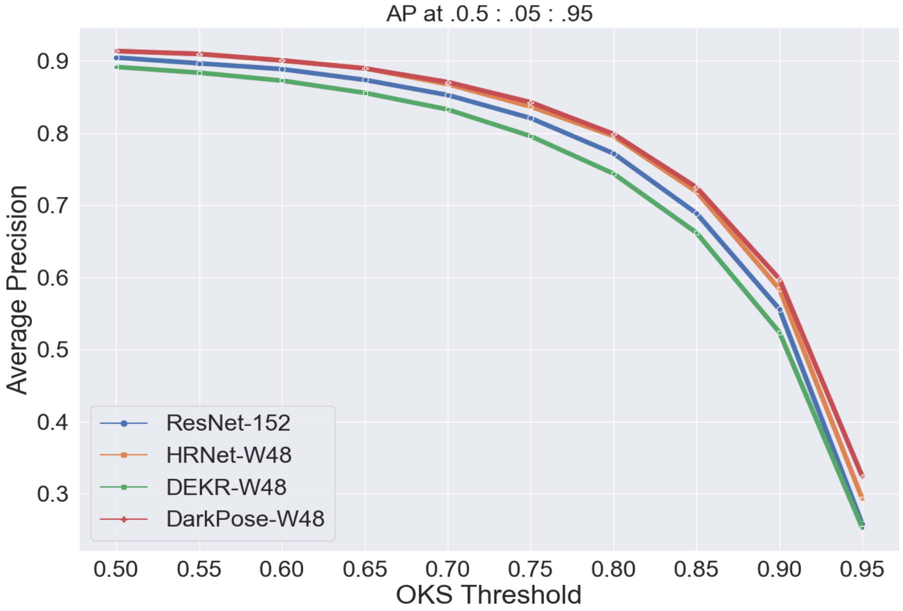


Figure 5.2: The AP score comparison for different OKS thresholds. The graph contains the AP at OKS thresholds: 0.5 until 0.95.

.

results. However, DEKR still performs similarly with AP^{50}, but from AP^{75} and onward the performance drops drastically. This can also be perceived in Figure 5.2 in which also can be seen that the scores at AP^{50} differ the least and become increasingly worse with a higher OKS threshold. Another observation is that HRNet and DarkPose reach exactly the same scores with the primary evaluation metric until an OKS threshold of .65. After this threshold DarkPose slightly outperforms HRNet.

Furthermore, to get a deeper insight into the results the AP and AR scores are also evaluated for the four groups of keypoints, namely the head, core body, arms and legs. The results of the four models with respect to these groups of keypoints can be found in Figure 5.3. Here it can be observed that the models are generally the best in predicting the keypoints belonging to the head and the worst at predicting the leg keypoints. For ResNet the performance of predicting the head keypoints is not better than for the core body keypoints. The same figure shows that the top-down models are generally better at correctly predicting the arms compared to the legs, while the performance of DEKR for arms and legs is relatively the same.

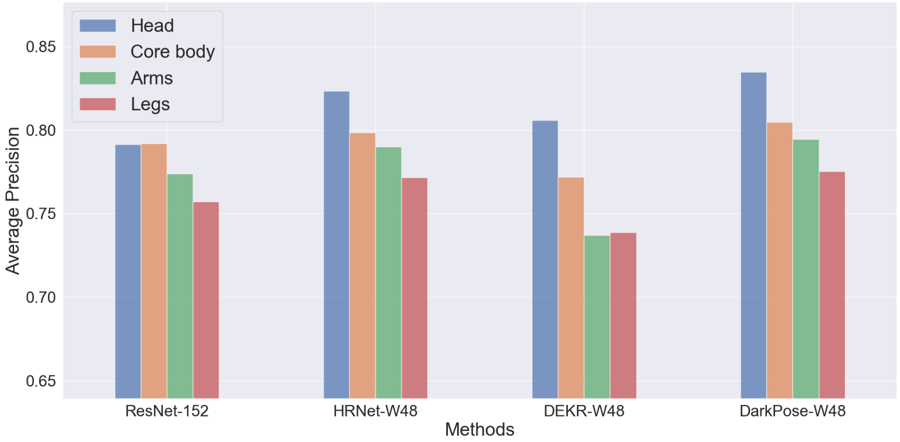


Figure 5.3: A comparison of the AP scores for the primary metric AP metric (at $OKS = .5 : 0.05 : .95$) on all labelled keypoints separated by the four keypoint groups: head, core body, arms and legs.

.

Method	Visibility	AP	AP^{50}	AP^{75}	AR	AR^{50}	AR^{75}
ResNet-152	Visible	0.794	0.915	0.856	0.847	0.955	0.901
	Occluded	0.554	0.803	0.577	0.666	0.878	0.694
HRNet-W48	Visible	0.811	0.922	0.869	0.859	0.956	0.908
	Occluded	0.576	0.825	0.611	0.685	0.891	0.722
DEKR-W48	Visible	0.776	0.902	0.839	0.829	0.945	0.883
	Occluded	0.530	0.782	0.553	0.637	0.853	0.665
DarkPose-W48	Visible	0.817	0.923	0.872	0.863	0.957	0.910
	Occluded	0.589	0.833	0.624	0.695	0.896	0.733

Table 5.2: A comparison of the performance on the visible and occluded keypoints from the newly annotated validation set of the COCO keypoint 2017 dataset.

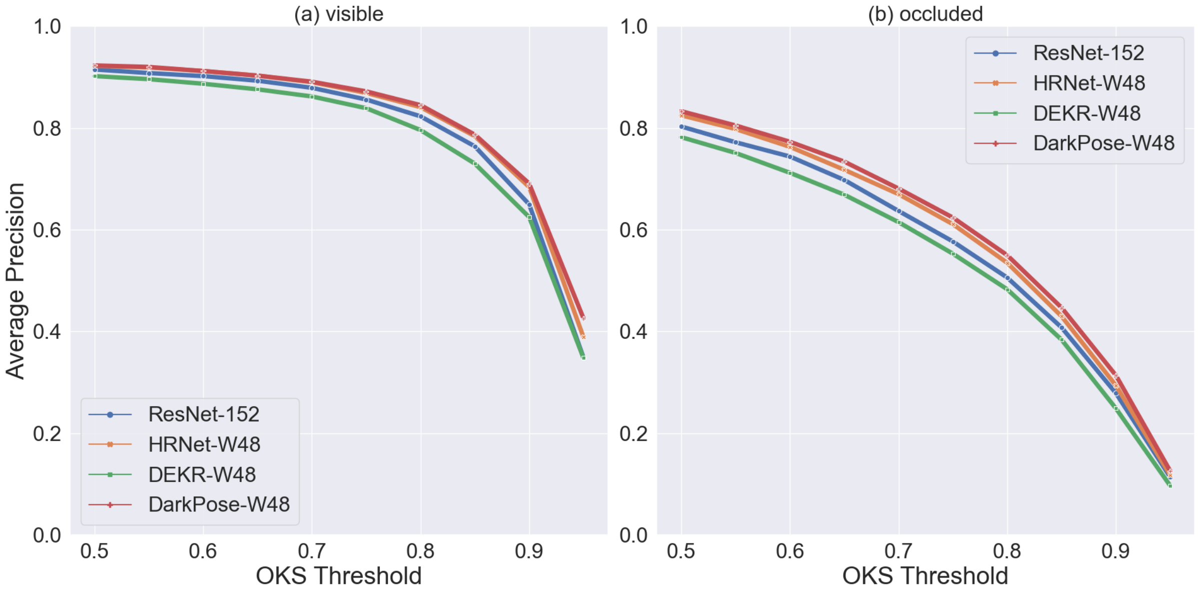


Figure 5.4: The AP scores for different OKS thresholds for the four models. The left graph contains the AP scores for the visible labels and the right with occlusion labels. Both tested with OKS thresholds: 0.5 until 0.95.

.

5.1.2 Occlusion

The models tend to follow the same pattern for visible and occluded keypoints as the general results. For both visible and occluded keypoints DarkPose-W48 performs the best and DEKR-W48 with multi-scaling performs the worst as Table 5.2 shows. Moreover, there is a large difference in performance when predicting visible keypoints compared to occluded keypoints. To get a better insight we will look at Figure 5.4. In this figure it can be perceived that the performance of all models on the visible keypoints tends to steadily drop until an OKS threshold of .9 and then drops drastically. The performance on the occluded keypoints is lower in general and the performance also drops relatively gradually.

The occluded keypoints have a strong negative influence on the performance of SOTA models with regards to AP and AR. This is in line with the literature [17, 27]. On the main evaluation metric the difference in performance between predicting the visible and occluded keypoints of every model drops on average 0.237 with a standard deviation of 0.007. With AP^{50} the performance only drops 0.105 on average with a standard deviation of 0.012. Hence, when the OKS threshold increases, the difference in AP scores for visible

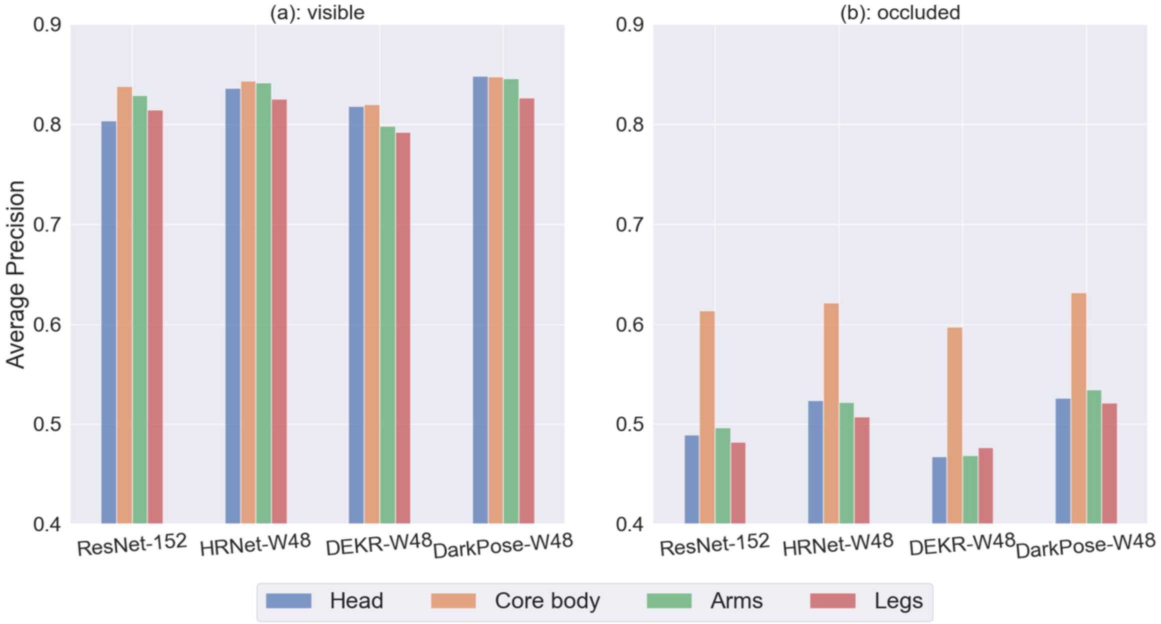


Figure 5.5: The AP score for every model for the four groups of keypoints where the left graph contains the results on solely the visible keypoints and the right the occluded keypoints.

and occluded labels tend to increase. What can also be noted here is that the difference between the scores on the visible and occluded labels is slightly but considerably smaller for DarkPose (0.228) than for ResNet (0.240) and DEKR (0.246). Additionally, the SOTA models maintain the same performance ranking on occluded labels and follow the same patterns as the results on the full validation set. Also, the performance on visible keypoints seems relatively stable until an OKS threshold of .8 and drops drastically after .9, when the room for error becomes possibly too small. This can also be seen in Figure 5.1, where the red targets are sometimes smaller than the actual keypoint (e.g. elbow and shoulder).

The performance of the models is also analyzed by the performance on the groups of keypoints. In Figure 5.5 can be observed that the models have a tendency to perform the best on the core body for both visible and occluded keypoints. Especially in the case of the occluded keypoints the performance on the core body keypoints is substantially higher than on the other keypoint groups. Also, all models seem to perform relatively well on the head keypoints when these are visible except for ResNet.

Method	Occlusion type	AP	AP^{50}	AP^{75}	AR	AR^{50}	AR^{75}
ResNet-152	Self	0.621	0.829	0.671	0.735	0.909	0.777
	Other person	0.405	0.619	0.403	0.595	0.812	0.608
	Environment	0.501	0.742	0.515	0.654	0.858	0.679
HRNet-W48	Self	0.636	0.849	0.690	0.745	0.915	0.790
	Other person	0.420	0.639	0.424	0.613	0.828	0.635
	Environment	0.538	0.771	0.566	0.683	0.877	0.718
DEKR-W48	Self	0.594	0.811	0.642	0.715	0.896	0.760
	Other person	0.380	0.599	0.374	0.536	0.766	0.542
	Environment	0.510	0.750	0.523	0.644	0.854	0.670
DarkPose-W48	Self	0.645	0.849	0.700	0.752	0.917	0.799
	Other person	0.428	0.650	0.440	0.628	0.842	0.664
	Environment	0.550	0.775	0.581	0.692	0.880	0.727

Table 5.3: Measurement of the performance of the models on the different occlusion types: self occlusion, other person occlusion and environment occlusion.

5.1.3 Occlusion Types

Occluded keypoints have a strong negative influence on the performance of SOTA models. Hence, the next step is to investigate whether the performance differs between different types of occlusion. That the results vary per type of occlusion is confirmed by the results shown in Table 5.3. Furthermore, the table also shows that all SOTA models perform the best on self occlusion and the worst on other person occlusion and that the differences between each of these occlusion types are substantial. The average difference in performance on these occlusion types can be observed in Table 5.4. This table confirms that the biggest difference in performance is between self occlusion and other person occlusion. And it also shows that there is a bigger performance difference between other person occlusion and environment occlusion (0.117), than between self occlusion and environment occlusion (0.099). Furthermore, Figure 5.6 shows that the performance of the models exponentially drops for self occlusion, less exponentially for environment occlusion and even less for other person occlusion.

AP differences occlusion types	Self occlusion	Other person occlusion	Environment occlusion
Self occlusion	0.000 (0.000)	0.216 (0.001)	0.099 (0.013)
Other person occlusion	0.216 (0.001)	0.000 (0.000)	0.117 (0.012)
Environment occlusion	0.099 (0.013)	0.117 (0.012)	0.000 (0.000)

Table 5.4: The differences in average performance on the primary evaluation metric between the different occlusion types with their corresponding standard deviations.

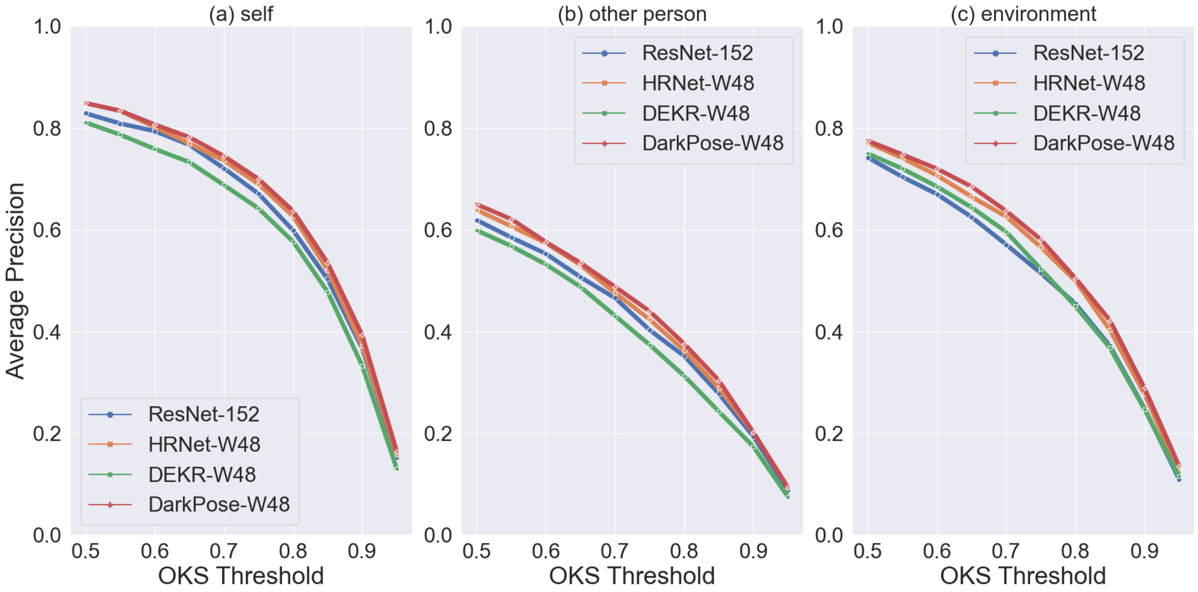


Figure 5.6: The AP scores for different OKS thresholds for the four models. The left graph contains the AP scores for the self occluded keypoints, the middle graph for the keypoints occluded by other persons and the right graph for the keypoints occluded by the environment. Both with OKS thresholds: 0.5 until 0.95.

.

Additionally, the models maintain the same ranking per occlusion type as the general results. So, every model performs the best on self occlusion, then the best on environment occlusion and the worst on other person occlusion. Hence, the occlusion types do not influence the models differently on first sight. When looking at Table 5.5, there can be observed that the performance difference follows the same order as well, except that the difference is smaller for ResNet. What can also be observed here is that there is only a relatively small performance difference between self and environment occlusion for DEKR compared to the other models, especially compared to ResNet.

Method	Self/ Environment	Self/ Other person	Environment/ Other person
ResNet-152	0.120	0.217	0.097
HRNet-W48	0.098	0.216	0.118
DarkPose-W48	0.095	0.216	0.121
DEKR-W48	0.084	0.214	0.130

Table 5.5: The differences in performance on the primary evaluation metric between the different occlusion types for every model.

5.1.4 Truncation at the Image Border

For truncation at the image border the difference in performance between truncation and no truncation was investigated, as well as the performance differences for varying numbers of truncated keypoints. So, Table 5.6 contains the differences in performance when truncation by the image border is present or not. And it shows that the performance drops substantially when at least one keypoint is truncated by the image border. When truncation is present the performance on the primary evaluation metric is on average 0.157 worse with a standard deviation of 0.029. It also follows the same ranking as for the other results. Darkpose performs the best, followed by HRNet and DEKR performs the worst. However, different from the occlusion labels, the performance drops exponentially when truncation is present as can be observed in Figure 5.7. This resembles the general results, and the results when occlusion and truncation are absent. The difference between absence and presence of truncation differs the most for DEKR (0.205), and is considerably lower for ResNet (0.150). Furthermore, the difference in AP scores of HRNet (0.135) and DarkPose (0.136) do not differ from each other, but are considerably lower than ResNet.

Furthermore, the performance gets worse when more keypoints are truncated by the image border as can be observed in Figure 5.8. When more keypoints are truncated the models are notably more likely to predict keypoints incorrectly, or are too far off when predicting the location. The differences between each following bin ($0 < k < 5$, $5 <= k < 9$, $9 <= k < 13$, $13 <= k$ respectively, where k is the number of keypoints truncated by the image border) differ less on average performance (0.251, 0.236, 0.167 respectively) with the primary evaluation metric. When looking at the performance between the different models it seems that ResNet generally has a considerably higher difference between the final two bins (0.194), while this difference is considerably lower for DEKR (0.107). Besides, normally AR is higher than AP, but the difference increases even more when more keypoints are truncated. When more keypoints are truncated the AP drops by a larger number than the AR. Hence, when more keypoints are truncated by the image border, the models are more likely to incorrectly predict the keypoint locations than to miss that the keypoints are present. This can be seen in Figure 5.8.

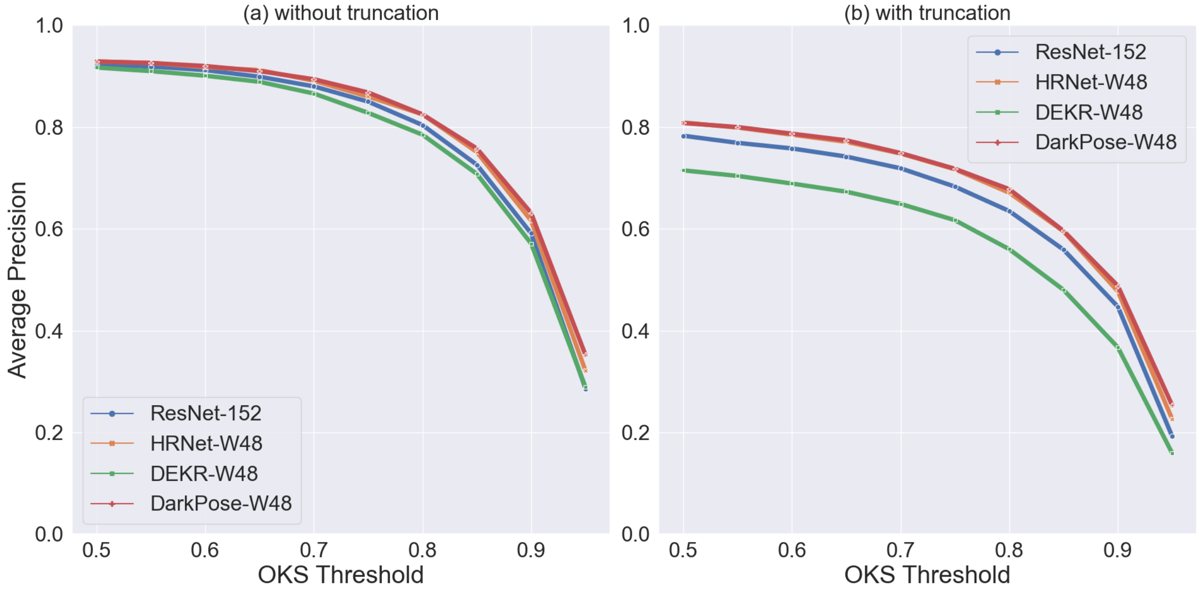


Figure 5.7: The AP scores for different OKS thresholds for four models. The left graph contains the AP scores for persons without any truncation and the right graph has the AP scores for people with at least one keypoint truncated by the image border. Both with OKS thresholds: 0.5 until 0.95.

.

Method	Truncation	AP	AP^{50}	AP^{75}	AR	AR^{50}	AR^{75}
ResNet-152	X	0.779	0.923	0.850	0.824	0.958	0.890
	V	0.629	0.783	0.683	0.767	0.919	0.821
HRNet-W48	X	0.795	0.929	0.860	0.837	0.960	0.899
	V	0.660	0.809	0.717	0.786	0.927	0.839
DEKR-W48	X	0.766	0.917	0.828	0.803	0.944	0.860
	V	0.561	0.715	0.617	0.752	0.915	0.808
DarkPose-W48	X	0.801	0.929	0.868	0.842	0.961	0.905
	V	0.665	0.808	0.718	0.790	0.926	0.841

Table 5.6: Results of the best performing models on the validation set of the COCO keypoint 2017 dataset with a split made for whether truncation is present.

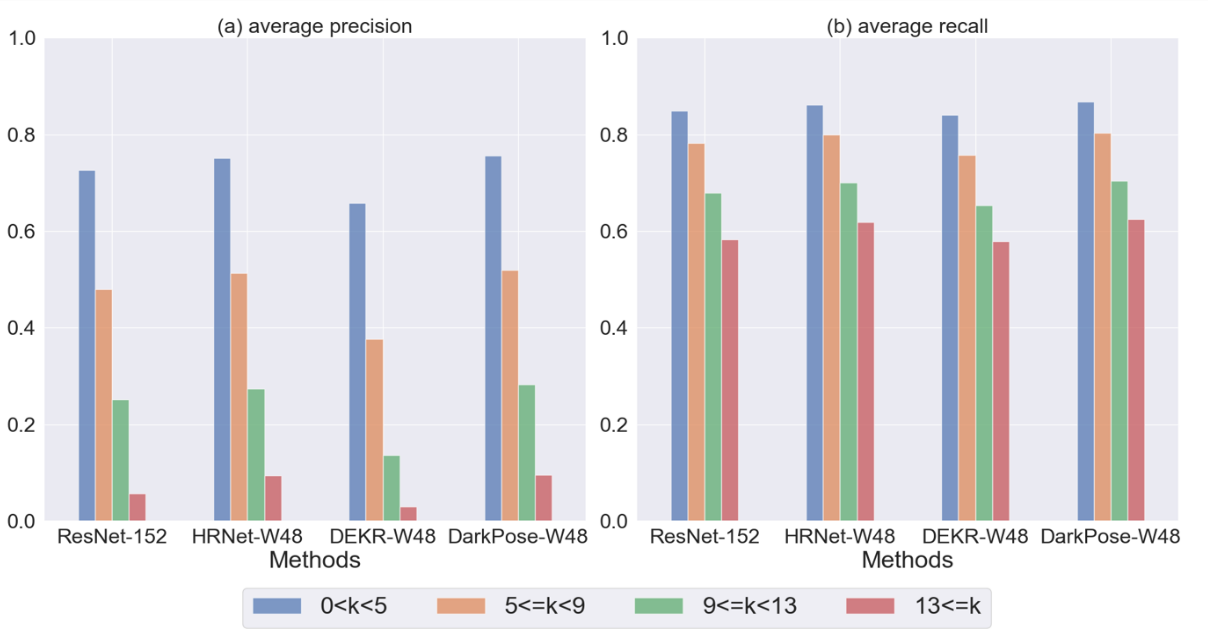


Figure 5.8: The AP and AR scores for every model for the four truncation bins where k is the number of keypoints truncated.

5.1.5 Image Resolution

To investigate the influence of the image resolution on the performance of SOTA models the image resolutions were categorized into three bins, based on the area of the bounding box for the corresponding person. These areas consist of the number of pixels in the automatically created bounding box. The three bins are named after the relative area r, namely: small ($r < 32^2$), medium ($32^2 <= r < 96^2$) and large ($92^2 <= r$). The results can be observed in Table 5.7. As the resolution of the person decreases, so does the performance of every model. The performance of the models in general does not differ that much for a large or medium person area, but drastically drops for a small area. The average difference in AP between the large and medium bin is 0.119 with a standard deviation of 0.009 and 0.483 with a standard deviation of 0.004 from a medium to a small bin. Furthermore, when comparing the different models there is one notable result. The AP difference between the large and medium bin differs considerably less for DEKR (0.103).

Method	Area	AP	AP^{50}	AP^{75}	AR	AR^{50}	AR^{75}
ResNet-152	Small	0.215	0.337	0.231	0.599	0.827	0.653
	Medium	0.695	0.859	0.766	0.771	0.929	0.839
	Large	0.817	0.937	0.882	0.874	0.981	0.931
HRNet-W48	Small	0.231	0.356	0.247	0.626	0.841	0.682
	Medium	0.714	0.872	0.784	0.784	0.932	0.852
	Large	0.837	0.946	0.896	0.889	0.984	0.939
DEKR-W48	Small	0.198	0.322	0.212	0.556	0.791	0.606
	Medium	0.687	0.863	0.755	0.747	0.920	0.811
	Large	0.790	0.905	0.847	0.864	0.971	0.913
DarkPose-W48	Small	0.238	0.361	0.253	0.630	0.846	0.691
	Medium	0.718	0.871	0.788	0.789	0.933	0.856
	Large	0.845	0.946	0.901	0.895	0.983	0.944

Table 5.7: The AP and AR scores for the models with a split made for the three person bounding box resolutions: small, medium and large.

5.1.6 Wrong Annotations

The wrong annotations do extensively influence the results as can be seen by observing the difference between the new and original results in Table 5.1. The influence of the wrong annotations can also be observed in Figure 5.9 in which the predictions by the models are shown on a person who is incorrectly labelled with the majority mistake: segmentation and bounding box do not correspond to the person keypoints. Here the OKS scores are very low for each keypoint, since the target keypoint for this person is mapped to another person. This effect of mistakes like this in the evaluation can also be observed in Table 5.8. What can be observed here is that the AP scores, as well as the AR are low. The AR is not influenced as much, since several keypoints are still predicted, but not at the correct area in the image.

Method	AP	AP^{50}	AP^{75}	AR	AR^{50}	AR^{75}
ResNet-152	0.167	0.290	0.158	0.493	0.729	0.495
HRNet-W48	0.185	0.319	0.188	0.512	0.748	0.551
DEKR-W48	0.211	0.364	0.189	0.463	0.720	0.449
DarkPose-W48	0.194	0.309	0.198	0.517	0.748	0.533

Table 5.8: Results on the 107 wrong annotations.

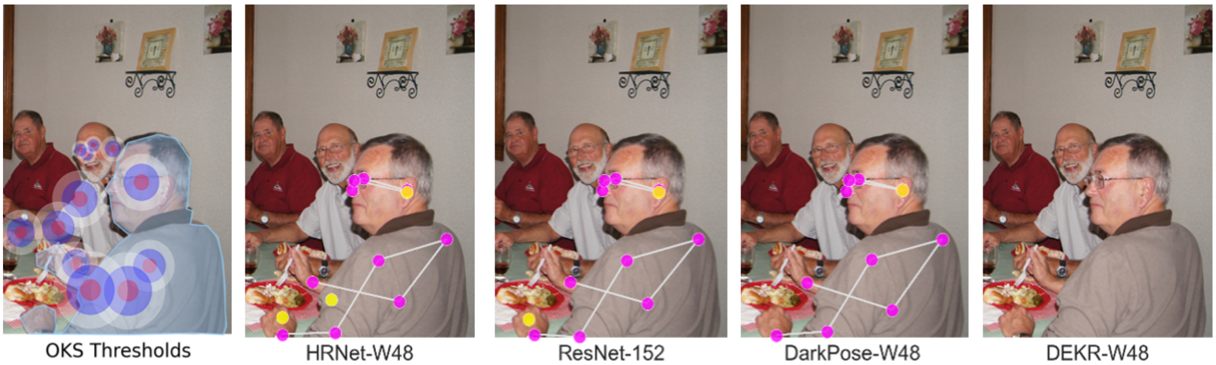


Figure 5.9: The most left image shows that the keypoints are not mapped to the segmented person. This image also contains the areas around the ground truth labels in which the models can obtain an OKS score of 0.5, 0.75 and 0.95 for white, blue and red respectively. Then to the right are the predictions of the models. In the images with predicted keypoints the colors correspond to the scores mapped to the OKS scores. The pink keypoints are mapped keypoints which do not get an OKS score of at least .5 and the yellow predicted keypoints do not affect the OKS score, since the unlabelled but predicted keypoints do not influence the score.

5.2 Filtering Method Results

The main idea of the filtering method was that removing wrong data instances from the train set might lead to better performance, since learning wrongly annotated data might lead to bias. Hence, the mistake that was made the majority of the time was analyzed and a method was found to filter more than 95% of the cases in which this mistake was made. Of the different versions able to filter at least this percentage, the version which also minimized the removal of correct annotations was chosen. The version which had the optimal balance between these two, of all versions, was described in Section 4.3. Then for every instance of the automatically filtered data was manually decided whether it should be removed. This was part of the process in order to minimize the removal of true labels from the train set. With the filtering method 874 of the 149.813 people annotated with at least one keypoint are removed from the train set (0.58%). The aim was to observe whether removing the wrong data instances would lead to better performance. For this we trained the pre-trained (on ImageNet) models once on the filtered data and once on the original data. The specific modes chosen are ResNet-50, HRNet-W32, DEKR-W32 and DarkPose-W32. All models use input size 256 × 192, except for DEKR which uses an input size of 512 × 512. Also, for DEKR multi-scale testing is used.

Method	Train version	AP	AP^{50}	AP^{75}	AR	AR^{50}	AR^{75}
ResNet-50	Filtered	0.711	0.895	0.791	0.771	0.937	0.842
	Original	0.709	0.895	0.786	0.769	0.937	0.839
HRNet-W32	Filtered	0.749	0.909	0.826	0.804	0.949	0.870
	Original	0.749	0.909	0.825	0.804	0.950	0.872
DEKR-W32	Filtered	0.714	0.888	0.781	0.768	0.949	0.880
	Original	0.707	0.884	0.770	0.760	0.923	0.816
DarkPose-W32	Filtered	0.764	0.909	0.833	0.817	0.949	0.880
	Original	0.765	0.910	0.834	0.814	0.948	0.875

Table 5.9: Results of the trained models. For each model one was trained on the filtered data and one on the original data.

The difference in performance between the models trained on the filtered and original data can be observed in Table 5.9. The table shows that the results are hardly influenced by filtering the train set. This is for multiple thresholds of AP as well as for AR. Only DEKR seems to profit slightly with a higher AP threshold and with different AR thresholds. However, since the differences on the averaged AP and AR are only small, the differences are deemed to be insubstantial. Hence, there seems to be no meaningful difference between the models trained on the filtered data and those trained on the original data. We also looked at occlusion and the occlusion types, since the filtered mistake always had wrong occlusion labels. Still, there is no difference in performance between the differently trained models with regards to these challenges.

Chapter 6

Discussion

In this chapter the results on the new annotations of the validation set of the COCO Keypoint Detection Task 2017 will be discussed separately and respectively for the new labels and the filtered data. We will end this chapter with the formulation of an answer for every research question.

6.1 New Annotations

The new annotations will be discussed in the same order as the presentation of the results, hence we will start with the general results, followed by the results on the specific contemporary challenges: occlusion, occlusion types, truncation by the image border, image resolution and wrong annotations.

The general results showed that the performance of all the tested models seems relatively stable until the prediction has to be localized more precisely. This might mean that performance with lower OKS threshold is close to a state of saturation, but there is still room for improvement for the higher thresholds. Still, what has to be regarded here is that from $OKS = 0.85$, human predictors tend to disagree on the location of the keypoints as well and their performance also drops drastically further on[1]. For the models the performance tends to drop exponentially with substantially worse performance from the same threshold. Furthermore, in general the top-down methods perform better than

[1] https://cocodataset.org/#keypoints-eval

the bottom-up method. This is in line with the literature [7]. The performance also differed when separating the keypoints of the head, core body, arms and legs. In general the models perform the best when predicting the keypoints belonging to the head, and the worst when predicting the legs. There are several likely explanations for the finding that the head keypoints are predicted the best. One is that the other keypoints have to deal more often with clothing which could negatively influence the prediction of these keypoints for several reasons on its own. Another is that the head keypoints have very distinguishable features, while keypoints such as elbows and knees resemble each other more. Also, the head keypoints have very similar shapes between instances, while the shapes of the other keypoint groups can vary more based on the body type. Finally, the most likely explanation is that challenging situations to predict the keypoints occur more often for the other keypoint groups. To sum up, the models were in general the best at predicting the keypoints belonging to the head, followed by those belonging to the core body. Additionally, there is no difference in the ordering of models with respect to performance with the new labels. Now the discussions will follow for each new annotation.

The results show a substantial difference in performance of SOTA models between predicting visible and occluded keypoints. Moreover, the performance drops gradually when the OKS threshold increases, instead of an exponential drop in performance for the visible keypoints. This does make sense, because the occluded keypoint cannot be observed directly so it has to be inferred. However, when inferring where the keypoint is, there is often the possibility that an occluded keypoint has multiple likely locations. And if the correct one is not guessed, it is less likely that the score is above the threshold. Also, a comparison of the performance of the models shows that the models follow the same pattern as the general results for both the visible and occluded labels. There is only a considerable deviation in the performance between the visible and occlusion labels when comparing DarkPose and ResNet. So, DarkPose seems to be more robust to occlusion than ResNet and DEKR. So, occlusion led to substantially worse performance for all models. And the performance of DarkPose is influenced less when faced with either a visible or occluded label compared to ResNet and DEKR.

Furthermore, the occlusion results also supported the claim that the models perform better when faced with the head keypoints. This is the case because in this dataset the head keypoints have to be less often predicted with challenging conditions than the other keypoint groups. This is supported by two factors. The first is that the performance on the head keypoints is for both conditions (i.e. visible and occluded) lower on average than the keypoints of the core body. The second is that in Section 4.2.2 was shown that the head keypoints are rarely occluded, compared to the other keypoint groups. Hence, it makes sense that the AP scores for the head are relatively high in general, but lower in both the visible and occlusion condition. Also, the average performance of the models on the core body keypoints is relatively good, especially in the occluded case. This might be because these keypoints are often in the approximately same position relative to the rest of the body. This can also be the case for the head, but it might be more difficult to get the same score, since the predictions have to be more precise for head keypoints. Hence, the models tend to generally perform the best on the core body, especially in the presence of occlusion.

To further investigate occlusion, several types of occlusion were distinguished: self occlusion, other person occlusion and environment occlusion. All of the different types negatively influence the performance of SOTA models. Other person occlusion seems to influence the performance the most and when it is present it leads to the worst results for every model when comparing the different types of occlusion. This is potentially due to the difficulty of separating the keypoints of two or more people, as this might lead the model to map the keypoints to the wrong person. After other person occlusion, environment occlusion leads to the worst results followed by self occlusion. Still it is important to consider self occlusion as well, since it still leads to a substantial performance drop compared to the general results. Furthermore, it is the case for every model that the performance with self occlusion is the best and with other person occlusion the worst. The difference between self occlusion and environment occlusion was relatively small for DEKR. Also, the difference was relativly big between environment occlusion and other person occlusion. This might indicate that DEKR handles environment occlusion relatively better compared to the other models. To conclude, it seems that the models tend to perform the best with self occlusion and the worst with other person occlusion.

Truncation by the image border was investigated by separating the people into bins, based on the number of keypoints truncated. Every following bin led to a substantial drop in performance. The results showed that the performance difference between presence and absence of truncation was substantially higher for DEKR followed by ResNet. So, it seems that presence of truncation leads to less problems during the prediction phase of better performing models. Furthermore, the performance drops less when the number of keypoints truncated by the image border increases, but in all cases it still leads to the performance dropping drastically. Here the performance difference between the last two bins is the smallest for DEKR and the highest for ResNet. So, it seems that DEKR is relatively worse at handling truncation in general, but when a high number of keypoints are truncated by the image border it matters less. While for ResNet it seems to be the case that it is not influenced more than the other models by the presence of truncation, but the performance suffers more when a high number of keypoints are truncated.

In a similar way as truncation by the image border the results of image resolution were also gathered using bins. These results follow the intuition that the more the resolution decreases, the lower the performance of the model becomes. Here the expectation was that models which use HRNet to at least extract the features from the image (i.e. HRNet, DEKR and DarkPose) would experience less performance drops with increasingly lower resolutions than ResNet. Since HRNet is designed to maintain a high resolution. Still, ResNet seems to have a similar decrease in performance as the other models when going to a bin with a smaller range of resolutions. The results also showed that the performance differences between the large and medium bin was the smallest for DEKR. This seems to indicate that the performance decreases the least for DEKR when the resolution decreases from high to medium.

The removal of the wrong annotations from the validation set led to higher results in general. The wrongly annotated data would lead to a flawed evaluation, even in the case where the predictions were correct. Still, the difference between the results on the filtered and the original dataset does not deviate between models.

So, in general every newly added annotation led to substantially worse performance. The inclusion of these annotations will provide a tool to get a better insight into the strengths and caveats of SOTA models in 2D multi person pose estimation.

6.2 Filtered Data

By filtering the data we investigated whether removing wrong data examples from the train set would influence the performance of SOTA models. First the general results were investigated, but there did not seem to be a performance difference between the models trained with and without filtered data. Then we also investigated the occlusion and occlusion type results, since every instance of the wrongly annotated data included wrong occlusion labelling. Nevertheless, for both results there was no substantial difference in performance for any of the models between their two versions. One expectation was that the performance of a simpler model such as ResNet-50 might improve more when trained on filtered data than on the original dataset. This was not the case, most likely since ResNet-50 is still a complex model and only relatively simple to the other models. Moreover, there were in general only minor differences in performance, but these can be attributed to slightly differing weight updates due to batches consisting of different images for both train sets. Still, it is still hard to conclude whether the removal of wrong data examples from the train set influences the performance of SOTA models.

On the one hand there are multiple reasons why it seems likely that the performance is not influenced by the removed data. One potential reason is that the removed data hardly influences the performance, since only 0.58% is removed. The amount of learning that happened might be nullified, because it is such a small percentage of the full train set. Another potential reason has to do with filtering just one type of mistake. This type of mistake, even though 76.64% of the mistakes made in this validation are of this type, might not influence the learning as much as the other types of mistakes in the train set.

On the other hand there are also reasons that filtering the data does enhance performance but cannot be observed from the data. The first is that the hyperparameters are not adjusted to the data. These are not learned from the data and have to be set before training to determine how the network is trained. Hence, there could have been a better configuration of hyperparameters for better performance for the models trained on the filtered data. Hence, with similar results this might mean the results are in favor of the models trained on the filtered data since their hyperparameters are not tuned. Another reason is that the models were still able to get nearly the same results with less data. To get nearly the same results with less data might mean that it performs better when the number of data instances would be equal. Even though it seems less likely that removing the wrongly annotated data from the train set influenced the performance, it cannot be ruled out.

In a nutshell, the filtered data did not have results necessarily supporting the claim that removing wrongly annotated data leads to better performance of SOTA models in 2D human pose estimation. These results have multiple explanations that support the claim, but these results also have explanations that oppose it.

6.3 Research Questions

In this section the research questions will be discussed. The main goal of this research was to investigate how SOTA models in 2D multi person pose estimation perform on challenges in the validation set of the COCO Keypoint Detection Task 2017. This leads us to answer the corresponding main and sub research questions here with the newly gained knowledge. Additionally, the research questions tied to the subsidiary research goal will be answered here. This subsidiary goal consisted of finding out whether removing wrongly annotated data from the train set would lead to better performance of SOTA models for the same validations set.

6.3.1 Sub-research question: How is the performance of state-of-the-art models affected by the specific challenges?

Occlusion types

Occlusion of keypoints in general had a substantial negative influence on the performance of SOTA models with the annotations. Moreover, when the occlusion labels are divided into the three sub-types: self occlusion, other person occlusion and environment occlusion, the results differ per type. All three types substantially decrease the performance, but other person occlusion decreases the performance the most, followed by environment occlusion and then by self occlusion.

Truncation by the image border

Truncation by the image border also influenced the results substantially. It follows a pattern such that the more keypoints are truncated by the image border, the worse the performance of the SOTA models becomes. Especially in the case of a large number of keypoints truncated by the image border, the performance suffers. Intuitively it seems that people will still perform well when a lot of truncation by the image border is present, but the models perform drastically low in this case.

Image resolution

Image resolution follows a similar pattern as truncation by the image border, that when the resolution of a person which has to be labelled decreases, the performance of the SOTA models will do that too. Especially when the resolution is low the performance suffers.

Wrong annotations

The wrong annotations also led to worse performance of the SOTA models for every model. The exclusion of the wrongly annotated data leads to a clearer picture of how the SOTA models perform.

6.3.2 Sub-research question: Do the state-of-the-art models differ in performance on these challenges?

The hypothesis was that the models would follow different performance patterns when faced with specific challenges. The state-of-the-art models perform differently on these challenges and the performance also responds differently to the degree specific challenges are present.

6.3.3 Main research question: How do state-of-the-art models in 2D multi person pose estimation perform on specific challenges present in the validation set of the COCO Keypoint Detection Task 2017

To answer the main research question it was divided into two sub-research questions. The answers to these two sub-research questions seem to indicate that the performance of state-of-the-art models drops based on the degree the challenge is present and the performance can differ per model when facing a certain challenge.

6.3.4 Subsidiary research question: Does removing the wrongly annotated data instances in the train phase lead to better performance?

To answer this research question the wrong data instances were automatically filtered and this filtered data was also manually checked to minimize filtering too many correct labels. It is also answered, solely for the COCO Keypoint Detection Task 2017. As discussed in Section 5.2, the performance in this setting does not seem to be influenced by removing the wrongly annotated data instances. However, it cannot be ruled out, since no hyperparameter tuning is performed and only one type of mistake is filtered.

Chapter 7

Limitations and Future Work

This study provides an insight into how the SOTA models perform with regards to the contemporary challenges in 2D human multi person pose estimation on the one hand. Besides, it gives insight into how one of the methods performs when wrong data is removed on the other hand. Both of these parts have their limitations, but also have promising future directions for upcoming research.

7.1 Limitations

For both the approach to the main and secondary research goal of this thesis there are corresponding limitations. First we will look at the limitations of the new labels and then for the removal of the labels. Finally, we will look at the limitations of both studies.

So, the first limitation for the new labels is that this study does not go into why one model performs better than the others. This is difficult in research in artificial intelligence in general, since architectures have complex differences which makes it difficult to pinpoint precisely to what extent a certain part of the architecture contributes to the difference in performance. However, in future work a better understanding of these different parts of the architectures, possibly with advances in the field of explainable AI, can contribute to a better view of where the field can advance.

Another limitation is the evaluation metric. The evaluation metric does measure labelled keypoint which are not predicted (with AR), but does not measure unlabelled keypoints which are predicted. A better representation of the true performance would include a measure of unlabelled but predicted keypoints. Moreover, this limitation is inherently tied to the COCO Keypoint Detection Task 2017, because not all keypoints are annotated in the image. An example of this can be observed in Figure 5.1 where the left hand is in the image and is unlabelled but predicted by three of the four models. Hence, labelling all keypoints in the image is required to include a measure like this. This would also lead to more data for training, validating and testing.

One more limitation is that the annotations could have been done more accurately. There are multiple factors which make it less accurate. One of these factors is that the resolution calculation depends on the number of keypoints annotated. It works better when more keypoints are annotated, since it is able to create a higher quality bounding box. However, with few keypoints annotated the bounding box might not correctly capture the visible parts of the person. This is partially alleviated by using a workaround when encountering minimal differences between the minimum and maximum on the x- and y-axis of the keypoint coordinates. Still, this might create bigger or smaller bounding boxes than the ideal bounding box. Another factor has to do with truncation by the image border. It works as it is supposed to, but it would have been better to get a deeper insight into this problem by labelling it per keypoint, instead of per person. By labelling it per keypoint the investigation might have gained more depth by looking into the performance differences for specific groups of keypoints being truncated.

Another limitation is that this research did not investigate the cases where multiple challenges are present at the same time. This was also hard to investigate with this dataset, since this led to very small sample sizes for conditions to investigate. This might be possible with a bigger dataset like AI Challenger.

For the removal of wrong instances the first limitation is that only a specific mistake is filtered. It greatly reduced the mistakes in the train set, but it is difficult to arrive at a conclusion of whether removal of the wrong data instances from the train set would lead to better performance, since not all of the wrong data was removed. Furthermore, tied to this limitation is also the limitation that the tool built to filter the data cannot be generalized to use for other datasets, hence the use is limited. Another limitation is that the models are not trained with tweaked hyperparameters on the filtered dataset. With tweaked hyperparameters the models might have been able to perform even better than now.

At last, the limitations for both research goals of this thesis are discussed. The first limitation here is that the data is annotated by one annotator. On the one hand this could also be an advantage, since the labelling is consistent, but on the other hand there are more downsides to it. Because there can be errors in the dataset, which is partially alleviated when multiple people annotate the same data and the best annotation for each annotation is used. The second limitation is that three of the four models rely on the HRNet backbone. This is also hard to avoid, since the SOTA methods of a certain time period usually use the same backbone which performs the best. Still, the performance of these models might resemble each other more, because of its reliance on the backbone.

7.2 Future Work

There are multiple directions for future work after this thesis for both research goals. We will first discuss the future directions for the new annotations and then for the filtered data.

For the new annotations, the test set of the COCO Keypoint Detection Task 2017 could also be annotated in future work to get an even better insight into the performance of the models. The models are generally optimized to perform as well as possible on the validation set, but might not generalize as well to the test set. Hence, to get a better insight it would be better to run the same experiment with the annotated test set and see whether the results follow the same pattern as those on the newly annotated validation set. For this the same annotation tool can be used as for this research. For us this was not possible, because the test set is kept private.

Also, future research could annotate similar benchmark datasets for keypoint detection with these labels to see whether the results are consistent when a dataset is used which emphasizes other challenges within human pose estimation specifically or computer vision in general. Additionally, whether the same conclusion can be inferred with a different evaluation metric. The disadvantage of the annotation tool is that it requires changes to apply to other datasets except for CrowdPose which uses a similar annotation method as COCO.

Another direction for future research is for the different types of occlusion to not only label the keypoint that is occluded, but also by what occludes. For self occlusion this might mean the keypoint which occludes, similarly as for other person occlusion, and the environment could use categories. This enables for example the investigation of the claim whether other person occlusion leads to the worst performance of the three types of occlusion due to mixing up the occluding person(s) and the occluded person(s).

Additionally, future research can focus on letting models predict the challenge it faces and for each challenge to what extent it is present (e.g. bin prediction for truncation by the image border and which type of occlusion for occlusion type). This can for example be used as additional input for a model that does human keypoint detection. To also predict to what extent the challenges are present would require labelling a bigger dataset with these labels first, since the train set of the COCO Keypoint Detection Task 2017 might not have enough instances of each challenge to properly learn to what degree the challenges are present.

Finally for the new annotations, the most straightforward future research is to use the newly annotated data to get a deeper insight into newly developed methods for the keypoint task of the COCO dataset.

For the filtered data, future work can also research whether training the models on the filtered train set will have other results with other hyperparameter configurations. This can be combined with training the models from scratch on a filtered version of ImageNet.

Chapter 8

Conclusion

To conclude the research of this thesis we looked at two topics in 2D multi person pose estimation. The first topic was how three types of occlusion (i.e. by self, another person or the environment), truncation at the image border, image resolution and wrong annotations influence the performance of state-of-the-art models and whether performance of these models differ on these challenges. The second topic was to investigate how the performance of the models changes when wrong annotations are removed from the train set.

These investigations led to the conclusions that (1) all label categories substantially influence the performance of state-of-the-art models and that the models performed differently when faced with the examined challenges. Furthermore, (2) the removal of wrongly annotated data from the train set seemed to not have influenced the performance of the models, but it requires more research to support this claim. Besides the analysis of these topics, we also provide publicly available annotations for other researchers to get a better insight into how their models perform on the validation set of the COCO Keypoint Detection Task 2017. These are available at `https://github.com/AgntBrwr/2d-human-pose-estimators-strenghts-and-caveats`. This provides possibilities for future research to get a deeper insight into where models lack and whether other models do differ in performance on the challenges in this dataset. On an ending note, the models shown in this thesis are still used for specific tasks, but are already less specific than earlier models. Hence, the question only arises when more general models will dominate this field.

Bibliography

[1] Andriluka, M., Pishchulin, L., Gehler, P., and Schiele, B. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis.

[2] Artacho, B., and Savakis, A. E. OmniPose: A Multi-Scale Framework for Multi-Person Pose Estimation. *CoRR abs/2103.10180* (2021).

[3] Bengio, Y., Simard, P., and Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks 5*, 2 (1994), 157–166.

[4] Cheng, B., Xiao, B., Wang, J., Shi, H., and Zhang, L. HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation. pp. 5385–5394.

[5] Cheng, Y., Yang, B., Wang, B., Wending, Y., and Tan, R. Occlusion-Aware Networks for 3D Human Pose Estimation in Video. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), pp. 723–732.

[6] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (2009), pp. 248–255.

[7] Geng, Z., Sun, K., Xiao, B., Zhang, Z., and Wang, J. Bottom-Up Human Pose Estimation Via Disentangled Keypoint Regression. In *CVPR* (2021).

[8] Hao, D., Zhang, L., Sumkin, J., Mohamed, A., and Wu, S. Inaccurate labels in weakly-supervised deep learning: Automatic identification and correction

and their impact on classification performance. *IEEE Journal of Biomedical and Health Informatics 24*, 9 (2020), 2701–2710.

[9] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778.

[10] Huang, J., Zhu, Z., Guo, F., and Huang, G. The Devil is in the Details: Delving into Unbiased Data Processing for Human Pose Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2020).

[11] Huang, J., Zhu, Z., Huang, G., and Du, D. AID: Pushing the Performance Boundary of Human Pose Estimation with Information Dropping Augmentation. *CoRR abs/2008.07139* (2020).

[12] Johnson, S., and Everingham, M. Learning effective human pose estimation from inaccurate annotation. In *CVPR 2011* (2011), pp. 1465–1472.

[13] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[14] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-Based Learning Applied to Document Recognition. In *Proceedings of the IEEE* (1998), vol. 86, pp. 2278–2324.

[15] LeCun, Y., and Cortes, C. MNIST handwritten digit database.

[16] Li, J., Wang, C., Zhu, H., Mao, Y., Fang, H.-S., and Lu, C. CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark. *arXiv preprint arXiv:1812.00324* (2018).

[17] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: Common Objects in Context.

In *Computer Vision – ECCV 2014* (Cham, 2014), D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Springer International Publishing, pp. 740–755.

[18] Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., and Bethge, M. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience 21*, 9 (2018), 1281–1289.

[19] McAuley, J., Targett, C., Shi, Q., and Hengel, A. v. d. Image-based recommendations on styles and substitutes, 2015.

[20] Moon, G., Chang, J. Y., and Lee, K. M. PoseFix: Model-agnostic General Human Pose Refinement Network, 2019.

[21] Newell, A., Yang, K., and Deng, J. Stacked Hourglass Networks for Human Pose Estimation. In *Computer Vision – ECCV 2016* (Cham, 2016), B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Springer International Publishing, pp. 483–499.

[22] Northcutt, C. G., Jiang, L., and Chuang, I. L. Confident Learning: Estimating Uncertainty in Dataset Labels, 2021.

[23] Padilla, R., Lobato Passos, W., Dias, T., Netto, S., and da Silva, E. A comparative analysis of object detection metrics with a companion open-source toolkit. *Electronics 10* (01 2021), 279–306.

[24] Papandreou, G., Zhu, T., Chen, L., Gidaris, S., Tompson, J., and Murphy, K. PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model. *CoRR abs/1803.08225* (2018).

[25] Papandreou, G., Zhu, T., Kanazawa, N., Toshev, A., Tompson, J., Bregler, C., and Murphy, K. P. Towards accurate multi-person pose estimation in the wild. *CoRR abs/1701.01779* (2017).

[26] Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence 39*, 6 (2017), 1137–1149.

[27] Ronchi, M. R., and Perona, P. Benchmarking and Error Diagnosis in Multi-Instance Pose Estimation, 2017.

[28] Song, H., Kim, M., Park, D., and Lee, J. Learning from noisy labels with deep neural networks: A survey. *CoRR abs/2007.08199* (2020).

[29] Sun, K., Xiao, B., Liu, D., and Wang, J. Deep High-Resolution Representation Learning for Human Pose Estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 5686–5696.

[30] Tompson, J., Jain, A., LeCun, Y., and Bregler, C. Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. In *NIPS* (2014).

[31] Toshev, A., and Szegedy, C. DeepPose: Human Pose Estimation via Deep Neural Networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 1653–1660.

[32] Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., and Xiao, B. Deep High-Resolution Representation Learning for Visual Recognition. *CoRR abs/1908.07919* (2019).

[33] Wang, K., Lin, L., Jiang, C., Qian, C., and Wei, P. 3D Human Pose Machines with Self-supervised Learning, 2019.

[34] Wu, J., Zheng, H., Zhao, B., Li, Y., Yan, B., Liang, R., Wang, W., Zhou, S., Lin, G., Fu, Y., and et al. Large-Scale Datasets for Going Deeper in Image Understanding. *2019 IEEE International Conference on Multimedia and Expo (ICME)* (2019).

[35] Xiao, B., Wu, H., and Wei, Y. Simple baselines for human pose estimation and tracking, 2018.

[36] Zhang, F., Zhu, X., Dai, H., Ye, M., and Zhu, C. Distribution-Aware Coordinate Representation for Human Pose Estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2020).

Appendix A

Full Results New Annotations

Note that the wrong annotations are filtered out of the validation set for all of the results below, hence the results are not the same as reported in the original papers. Also, in all of these tables for the DEKR model ms means that multi-scale testing is used.

Method	Input size	AP	AP^{50}	AP^{75}	AR	AR^{50}	AR^{75}
ResNet-50	256 × 192	0.712	0.895	0.792	0.772	0.937	0.843
ResNet-101	256 × 192	0.723	0.902	0.804	0.780	0.943	0.851
ResNet-152	256 × 192	0.729	0.902	0.808	0.787	0.942	0.857
ResNet-152	384 × 288	0.752	0.905	0.821	0.805	0.945	0.868
HRNet-W32	256 × 192	0.752	0.912	0.829	0.807	0.950	0.876
HRNet-W32	384 × 288	0.766	0.914	0.835	0.818	0.950	0.879
HRNet-W48	256 × 192	0.758	0.912	0.833	0.811	0.948	0.876
HRNet-W48	384 × 288	0.771	0.914	0.837	0.820	0.949	0.879
DEKR-W32 ms	512 × 512	0.715	0.887	0.781	0.767	0.926	0.824
DEKR-W32	512 × 512	0.688	0.877	0.754	0.739	0.909	0.793
DEKR-W48 ms	640 × 640	0.732	0.892	0.796	0.786	0.934	0.843
DEKR-W48	640 × 640	0.718	0.892	0.782	0.770	0.925	0.824
DarkPose-W32	128 × 96	0.715	0.897	0.793	0.775	0.940	0.843
DarkPose-W32	256 × 192	0.765	0.913	0.832	0.816	0.951	0.876
DarkPose-W32	384 × 288	0.775	0.916	0.837	0.824	0.951	0.880
DarkPose-W48	128 × 96	0.726	0.900	0.805	0.785	0.941	0.855
DarkPose-W48	384 × 288	0.777	0.914	0.843	0.825	0.949	0.884

Table A.1: Results of all the tested methods on the validation set of theCOCO Keypoint Detection Task 2017.

Method	Input size	Visibility	AP	AP^{50}	AP^{75}	AR	AR^{50}	AR^{75}
ResNet-50	256 × 192	Visible	0.756	0.906	0.832	0.816	0.948	0.881
ResNet-50	256 × 192	Occluded	0.501	0.763	0.520	0.629	0.856	0.658
ResNet-101	256 × 192	Visible	0.764	0.908	0.839	0.822	0.950	0.885
ResNet-101	256 × 192	Occluded	0.520	0.782	0.543	0.646	0.871	0.677
ResNet-152	256 × 192	Visible	0.770	0.912	0.845	0.828	0.952	0.891
ResNet-152	256 × 192	Occluded	0.528	0.791	0.550	0.649	0.874	0.680
ResNet-152	384 × 288	Visible	0.794	0.915	0.856	0.847	0.955	0.901
ResNet-152	384 × 288	Occluded	0.554	0.803	0.577	0.666	0.878	0.694
HRNet-W32	256 × 192	Visible	0.794	0.920	0.863	0.848	0.958	0.907
HRNet-W32	256 × 192	Occluded	0.543	0.801	0.569	0.666	0.882	0.700
HRNet-W32	384 × 288	Visible	0.808	0.921	0.866	0.858	0.958	0.908
HRNet-W32	384 × 288	Occluded	0.564	0.817	0.597	0.680	0.891	0.719
HRNet-W48	256 × 192	Visible	0.800	0.921	0.866	0.851	0.957	0.907
HRNet-W48	256 × 192	Occluded	0.561	0.813	0.596	0.678	0.886	0.718
HRNet-W48	384 × 288	Visible	0.811	0.922	0.869	0.859	0.956	0.908
HRNet-W48	384 × 288	Occluded	0.576	0.825	0.611	0.685	0.891	0.722
DEKR-W32 ms	512 × 512	Visible	0.758	0.897	0.816	0.809	0.935	0.857
DEKR-W32 ms	512 × 512	Occluded	0.512	0.768	0.528	0.619	0.840	0.641
DEKR-W32	512 × 512	Visible	0.729	0.886	0.791	0.779	0.918	0.830
DEKR-W32	512 × 512	Occluded	0.494	0.751	0.508	0.594	0.819	0.615
DEKR-W48 ms	640 × 640	Visible	0.776	0.902	0.839	0.829	0.945	0.883
DEKR-W48 ms	640 × 640	Occluded	0.530	0.782	0.553	0.637	0.853	0.665
DEKR-W48	640 × 640	Visible	0.762	0.902	0.826	0.811	0.935	0.862
DEKR-W48	640 × 640	Occluded	0.526	0.782	0.546	0.621	0.841	0.648
DarkPose-W32	128 × 96	Visible	0.758	0.906	0.834	0.817	0.948	0.881
DarkPose-W32	128 × 96	Occluded	0.494	0.757	0.507	0.631	0.859	0.656
DarkPose-W32	256 × 192	Visible	0.806	0.919	0.864	0.856	0.958	0.905
DarkPose-W32	256 × 192	Occluded	0.564	0.810	0.592	0.677	0.887	0.711
DarkPose-W32	384 × 288	Visible	0.816	0.922	0.868	0.862	0.957	0.908
DarkPose-W32	384 × 288	Occluded	0.578	0.821	0.603	0.689	0.892	0.722
DarkPose-W48	128 × 96	Visible	0.767	0.906	0.840	0.825	0.948	0.888
DarkPose-W48	128 × 96	Occluded	0.517	0.781	0.536	0.647	0.871	0.679
DarkPose-W48	384 × 288	Visible	0.817	0.923	0.872	0.863	0.957	0.910
DarkPose-W48	384 × 288	Occluded	0.589	0.833	0.624	0.695	0.896	0.733

Table A.2: Results of all the tested methods on visible and occluded keypoints on the validation set of the COCO Keypoint Detection Task 2017.

Method	Input size	Occlusion Type	AP	AP^{50}	AP^{75}	AR	AR^{50}	AR^{75}
ResNet-50	256 × 192	Self	0.566	0.790	0.604	0.695	0.883	0.737
ResNet-50	256 × 192	Person	0.357	0.566	0.353	0.563	0.782	0.578
ResNet-50	256 × 192	Environment	0.450	0.696	0.456	0.614	0.834	0.634
ResNet-101	256 × 192	Self	0.573	0.791	0.619	0.704	0.887	0.751
ResNet-101	256 × 192	Person	0.371	0.588	0.373	0.580	0.809	0.596
ResNet-101	256 × 192	Environment	0.478	0.721	0.491	0.643	0.853	0.672
ResNet-152	256 × 192	Self	0.591	0.817	0.630	0.714	0.902	0.757
ResNet-152	256 × 192	Person	0.377	0.590	0.373	0.578	0.798	0.590
ResNet-152	256 × 192	Environment	0.481	0.724	0.503	0.643	0.853	0.676
ResNet-152	384 × 288	Self	0.621	0.829	0.671	0.735	0.909	0.777
ResNet-152	384 × 288	Person	0.405	0.619	0.403	0.595	0.812	0.608
ResNet-152	384 × 288	Environment	0.501	0.742	0.515	0.654	0.858	0.679
HRNet-W32	256 × 192	Self	0.601	0.816	0.653	0.723	0.897	0.772
HRNet-W32	256 × 192	Person	0.403	0.631	0.392	0.607	0.832	0.617
HRNet-W32	256 × 192	Environment	0.501	0.749	0.517	0.661	0.867	0.694
HRNet-W32	384 × 288	Self	0.614	0.836	0.664	0.734	0.912	0.781
HRNet-W32	384 × 288	Person	0.410	0.631	0.418	0.612	0.828	0.635
HRNet-W32	384 × 288	Environment	0.533	0.768	0.565	0.679	0.877	0.718
HRNet-W48	256 × 192	Self	0.616	0.830	0.665	0.733	0.904	0.777
HRNet-W48	256 × 192	Person	0.412	0.627	0.416	0.614	0.830	0.635
HRNet-W48	256 × 192	Environment	0.523	0.766	0.548	0.676	0.877	0.713
HRNet-W48	384 × 288	Self	0.636	0.849	0.690	0.745	0.915	0.790
HRNet-W48	384 × 288	Person	0.420	0.639	0.424	0.613	0.828	0.635
HRNet-W48	384 × 288	Environment	0.538	0.771	0.566	0.683	0.877	0.718
DEKR-W32 ms	512 × 512	Self	0.575	0.801	0.613	0.694	0.886	0.730
DEKR-W32 ms	512 × 512	Person	0.372	0.594	0.367	0.524	0.751	0.533
DEKR-W32 ms	512 × 512	Environment	0.485	0.725	0.498	0.623	0.839	0.645
DEKR-W32	512 × 512	Self	0.558	0.785	0.599	0.667	0.862	0.707
DEKR-W32	512 × 512	Person	0.356	0.571	0.349	0.501	0.721	0.507
DEKR-W32	512 × 512	Environment	0.462	0.708	0.468	0.592	0.812	0.612
DEKR-W48 ms	640 × 640	Self	0.594	0.811	0.642	0.715	0.896	0.760
DEKR-W48 ms	640 × 640	Person	0.380	0.599	0.374	0.536	0.766	0.542
DEKR-W48 ms	640 × 640	Environment	0.510	0.750	0.523	0.644	0.854	0.670
DEKR-W48	640 × 640	Self	0.583	0.801	0.626	0.694	0.878	0.733
DEKR-W48	640 × 640	Person	0.390	0.619	0.382	0.528	0.760	0.535
DEKR-W48	640 × 640	Environment	0.499	0.744	0.510	0.625	0.837	0.649
DarkPose-W32	128 × 96	Self	0.553	0.781	0.592	0.688	0.881	0.733
DarkPose-W32	128 × 96	Person	0.359	0.579	0.352	0.573	0.799	0.582
DarkPose-W32	128 × 96	Environment	0.451	0.700	0.462	0.628	0.843	0.655
DarkPose-W32	256 × 192	Self	0.625	0.826	0.676	0.740	0.906	0.786
DarkPose-W32	256 × 192	Person	0.412	0.623	0.417	0.614	0.821	0.636
DarkPose-W32	256 × 192	Environment	0.517	0.752	0.538	0.669	0.872	0.702
DarkPose-W32	384 × 288	Self	0.634	0.841	0.675	0.747	0.916	0.789
DarkPose-W32	384 × 288	Person	0.421	0.637	0.421	0.623	0.832	0.640
DarkPose-W32	384 × 288	Environment	0.539	0.765	0.566	0.687	0.878	0.720
DarkPose-W48	128 × 96	Self	0.576	0.797	0.617	0.706	0.892	0.751
DarkPose-W48	128 × 96	Person	0.381	0.604	0.377	0.591	0.821	0.615
DarkPose-W48	128 × 96	Environment	0.470	0.724	0.478	0.639	0.856	0.664
DarkPose-W48	384 × 288	Self	0.645	0.849	0.700	0.752	0.917	0.799
DarkPose-W48	384 × 288	Person	0.428	0.650	0.440	0.628	0.842	0.664
DarkPose-W48	384 × 288	Environment	0.550	0.775	0.581	0.692	0.880	0.727

Table A.3: Results of all the tested methods for the different occlusion types on the validation set of the COCO Keypoint Detection Task 2017.

Method	Input size	k	AP	AP^{50}	AP^{75}	AR	AR^{50}	AR^{75}
ResNet-50	256 × 192	$k = 0$	0.742	0.916	0.822	0.791	0.953	0.867
ResNet-50	256 × 192	$0 < k$	0.585	0.759	0.652	0.732	0.905	0.794
ResNet-101	256 × 192	$k = 0$	0.751	0.920	0.836	0.800	0.958	0.876
ResNet-101	256 × 192	$0 < k$	0.595	0.771	0.659	0.740	0.913	0.799
ResNet-152	256 × 192	$k = 0$	0.756	0.919	0.839	0.805	0.956	0.881
ResNet-152	256 × 192	$0 < k$	0.605	0.772	0.667	0.749	0.915	0.807
ResNet-152	384 × 288	$k = 0$	0.779	0.923	0.850	0.824	0.958	0.890
ResNet-152	384 × 288	$0 < k$	0.629	0.783	0.683	0.767	0.919	0.821
HRNet-W32	256 × 192	$k = 0$	0.775	0.929	0.853	0.822	0.962	0.895
HRNet-W32	256 × 192	$0 < k$	0.643	0.801	0.709	0.776	0.926	0.837
HRNet-W32	384 × 288	$k = 0$	0.789	0.926	0.859	0.832	0.959	0.898
HRNet-W32	384 × 288	$0 < k$	0.657	0.806	0.715	0.789	0.931	0.841
HRNet-W48	256 × 192	$k = 0$	0.782	0.926	0.858	0.827	0.959	0.895
HRNet-W48	256 × 192	$0 < k$	0.646	0.803	0.713	0.778	0.925	0.836
HRNet-W48	384 × 288	$k = 0$	0.795	0.929	0.860	0.837	0.960	0.899
HRNet-W48	384 × 288	$0 < k$	0.660	0.809	0.717	0.786	0.927	0.839
DEKR-W32 ms	512 × 512	$k = 0$	0.747	0.911	0.816	0.784	0.937	0.844
DEKR-W32 ms	512 × 512	$0 < k$	0.558	0.720	0.612	0.732	0.902	0.783
DEKR-W32	512 × 512	$k = 0$	0.717	0.902	0.787	0.753	0.923	0.815
DEKR-W32	512 × 512	$0 < k$	0.541	0.714	0.585	0.707	0.878	0.748
DEKR-W48 ms	640 × 640	$k = 0$	0.766	0.917	0.828	0.803	0.944	0.860
DEKR-W48 ms	640 × 640	$0 < k$	0.561	0.715	0.617	0.752	0.915	0.808
DEKR-W48	640 × 640	$k = 0$	0.753	0.918	0.817	0.788	0.940	0.843
DEKR-W48	640 × 640	$0 < k$	0.550	0.716	0.602	0.732	0.894	0.784
DarkPose-W32	128 × 96	$k = 0$	0.741	0.917	0.822	0.794	0.954	0.867
DarkPose-W32	128 × 96	$0 < k$	0.595	0.770	0.658	0.735	0.911	0.792
DarkPose-W32	256 × 192	$k = 0$	0.788	0.927	0.857	0.831	0.961	0.896
DarkPose-W32	256 × 192	$0 < k$	0.652	0.800	0.707	0.785	0.930	0.835
DarkPose-W32	384 × 288	$k = 0$	0.798	0.932	0.861	0.840	0.962	0.899
DarkPose-W32	384 × 288	$0 < k$	0.659	0.801	0.709	0.792	0.928	0.840
DarkPose-W48	128 × 96	$k = 0$	0.753	0.917	0.837	0.804	0.954	0.880
DarkPose-W48	128 × 96	$0 < k$	0.602	0.770	0.660	0.744	0.914	0.803
DarkPose-W48	384 × 288	$k = 0$	0.801	0.929	0.868	0.842	0.961	0.905
DarkPose-W48	384 × 288	$0 < k$	0.665	0.808	0.718	0.790	0.926	0.841

Table A.4: Results of all the tested methods for presence or absence of truncation by the image border on the validation set of the COCO Keypoint Detection Task 2017.

Method	Input size	k	AP	AP^{50}	AP^{75}	AR	AR^{50}	AR^{75}
ResNet-50	256×192	$0 < k < 5$	0.679	0.821	0.756	0.813	0.955	0.888
ResNet-50	256×192	$5 <= k < 9$	0.434	0.592	0.490	0.752	0.943	0.826
ResNet-50	256×192	$9 <= k < 13$	0.215	0.314	0.233	0.644	0.830	0.691
ResNet-50	256×192	$13 <= k$	0.046	0.083	0.044	0.541	0.765	0.558
ResNet-101	256×192	$0 < k < 5$	0.690	0.833	0.764	0.822	0.961	0.893
ResNet-101	256×192	$5 <= k < 9$	0.441	0.598	0.493	0.759	0.953	0.828
ResNet-101	256×192	$9 <= k < 13$	0.220	0.317	0.238	0.655	0.835	0.694
ResNet-101	256×192	$13 <= k$	0.048	0.092	0.048	0.541	0.781	0.570
ResNet-152	256×192	$0 < k < 5$	0.702	0.837	0.773	0.831	0.963	0.906
ResNet-152	256×192	$5 <= k < 9$	0.438	0.588	0.485	0.764	0.945	0.836
ResNet-152	256×192	$9 <= k < 13$	0.223	0.319	0.243	0.655	0.843	0.683
ResNet-152	256×192	$13 <= k$	0.053	0.096	0.051	0.569	0.797	0.586
ResNet-152	384×288	$0 < k < 5$	0.726	0.846	0.786	0.849	0.962	0.906
ResNet-152	384×288	$5 <= k < 9$	0.480	0.623	0.523	0.782	0.949	0.844
ResNet-152	384×288	$9 <= k < 13$	0.252	0.351	0.272	0.679	0.853	0.723
ResNet-152	384×288	$13 <= k$	0.058	0.104	0.056	0.582	0.805	0.625
HRNet-W32	256×192	$0 < k < 5$	0.731	0.862	0.806	0.845	0.966	0.912
HRNet-W32	256×192	$5 <= k < 9$	0.509	0.655	0.563	0.794	0.953	0.868
HRNet-W32	256×192	$9 <= k < 13$	0.289	0.391	0.317	0.691	0.859	0.738
HRNet-W32	256×192	$13 <= k$	0.073	0.127	0.071	0.626	0.833	0.661
HRNet-W32	384×288	$0 < k < 5$	0.745	0.863	0.809	0.860	0.967	0.919
HRNet-W32	384×288	$5 <= k < 9$	0.513	0.656	0.560	0.807	0.958	0.868
HRNet-W32	384×288	$9 <= k < 13$	0.290	0.395	0.313	0.709	0.874	0.749
HRNet-W32	384×288	$13 <= k$	0.076	0.131	0.072	0.621	0.833	0.657
HRNet-W48	256×192	$0 < k < 5$	0.738	0.866	0.814	0.851	0.967	0.919
HRNet-W48	256×192	$5 <= k < 9$	0.496	0.642	0.549	0.794	0.951	0.850
HRNet-W48	256×192	$9 <= k < 13$	0.270	0.376	0.293	0.699	0.866	0.757
HRNet-W48	256×192	$13 <= k$	0.079	0.130	0.076	0.609	0.809	0.633
HRNet-W48	384×288	$0 < k < 5$	0.751	0.862	0.815	0.862	0.967	0.922
HRNet-W48	384×288	$5 <= k < 9$	0.513	0.661	0.563	0.800	0.949	0.862
HRNet-W48	384×288	$9 <= k < 13$	0.275	0.382	0.290	0.701	0.861	0.743
HRNet-W48	384×288	$13 <= k$	0.094	0.156	0.093	0.619	0.837	0.645
DEKR-W32 ms	512×512	$0 < k < 5$	0.672	0.807	0.736	0.827	0.956	0.891
DEKR-W32 ms	512×512	$5 <= k < 9$	0.383	0.534	0.423	0.744	0.927	0.802
DEKR-W32 ms	512×512	$9 <= k < 13$	0.134	0.203	0.143	0.607	0.801	0.644
DEKR-W32 ms	512×512	$13 <= k$	0.030	0.055	0.028	0.563	0.813	0.574
DEKR-W32	512×512	$0 < k < 5$	0.634	0.778	0.695	0.818	0.953	0.880
DEKR-W32	512×512	$5 <= k < 9$	0.366	0.533	0.377	0.723	0.915	0.757
DEKR-W32	512×512	$9 <= k < 13$	0.164	0.254	0.174	0.576	0.775	0.594
DEKR-W32	512×512	$13 <= k$	0.045	0.081	0.044	0.484	0.701	0.494
DEKR-W48 ms	640×640	$0 < k < 5$	0.659	0.788	0.715	0.841	0.961	0.898
DEKR-W48 ms	640×640	$5 <= k < 9$	0.377	0.525	0.424	0.758	0.943	0.820
DEKR-W48 ms	640×640	$9 <= k < 13$	0.137	0.203	0.146	0.654	0.851	0.704
DEKR-W48 ms	640×640	$13 <= k$	0.030	0.050	0.032	0.579	0.801	0.625
DEKR-W48	640×640	$0 < k < 5$	0.636	0.771	0.691	0.832	0.958	0.888
DEKR-W48	640×640	$5 <= k < 9$	0.366	0.522	0.406	0.738	0.919	0.802
DEKR-W48	640×640	$9 <= k < 13$	0.159	0.239	0.166	0.623	0.804	0.660
DEKR-W48	640×640	$13 <= k$	0.044	0.077	0.046	0.529	0.753	0.566
DarkPose-W32	128×96	$0 < k < 5$	0.686	0.836	0.763	0.813	0.961	0.883
DarkPose-W32	128×96	$5 <= k < 9$	0.445	0.605	0.488	0.758	0.947	0.824
DarkPose-W32	128×96	$9 <= k < 13$	0.242	0.339	0.263	0.652	0.846	0.686
DarkPose-W32	128×96	$13 <= k$	0.049	0.090	0.049	0.541	0.761	0.570
DarkPose-W32	256×192	$0 < k < 5$	0.741	0.852	0.805	0.860	0.966	0.920
DarkPose-W32	256×192	$5 <= k < 9$	0.504	0.642	0.542	0.797	0.949	0.852
DarkPose-W32	256×192	$9 <= k < 13$	0.279	0.380	0.299	0.709	0.885	0.757
DarkPose-W32	256×192	$13 <= k$	0.072	0.123	0.071	0.613	0.837	0.618
DarkPose-W32	384×288	$0 < k < 5$	0.747	0.855	0.801	0.867	0.965	0.916
DarkPose-W32	384×288	$5 <= k < 9$	0.496	0.637	0.537	0.809	0.955	0.864
DarkPose-W32	384×288	$9 <= k < 13$	0.269	0.368	0.283	0.710	0.866	0.746
DarkPose-W32	384×288	$13 <= k$	0.086	0.139	0.088	0.619	0.833	0.665
DarkPose-W48	128×96	$0 < k < 5$	0.689	0.829	0.756	0.822	0.962	0.894
DarkPose-W48	128×96	$5 <= k < 9$	0.453	0.608	0.497	0.763	0.945	0.832
DarkPose-W48	128×96	$9 <= k < 13$	0.244	0.335	0.263	0.657	0.835	0.696
DarkPose-W48	128×96	$13 <= k$	0.051	0.095	0.051	0.560	0.805	0.582
DarkPose-W48	384×288	$0 < k < 5$	0.757	0.861	0.816	0.868	0.965	0.924
DarkPose-W48	384×288	$5 <= k < 9$	0.519	0.660	0.563	0.803	0.947	0.854
DarkPose-W48	384×288	$9 <= k < 13$	0.283	0.387	0.299	0.704	0.861	0.751
DarkPose-W48	384×288	$13 <= k$	0.095	0.155	0.094	0.624	0.845	0.661

Table A.5: Results of all the tested methods for different types of truncation on the validation set of the COCO Keypoint Detection Task 2017.

Method	Input size	Area	AP	AP^{50}	AP^{75}	AR	AR^{50}	AR^{75}
ResNet-50	256 × 192	Small	0.193	0.308	0.213	0.580	0.819	0.639
ResNet-50	256 × 192	Medium	0.660	0.850	0.733	0.739	0.921	0.812
ResNet-50	256 × 192	Large	0.771	0.924	0.854	0.836	0.973	0.909
ResNet-101	256 × 192	Small	0.198	0.316	0.213	0.596	0.841	0.651
ResNet-101	256 × 192	Medium	0.668	0.854	0.745	0.747	0.926	0.819
ResNet-101	256 × 192	Large	0.781	0.931	0.866	0.845	0.978	0.917
ResNet-152	256 × 192	Small	0.201	0.320	0.211	0.598	0.831	0.641
ResNet-152	256 × 192	Medium	0.673	0.851	0.751	0.754	0.927	0.830
ResNet-152	256 × 192	Large	0.789	0.932	0.869	0.851	0.976	0.920
ResNet-152	384 × 288	Small	0.215	0.337	0.231	0.599	0.827	0.653
ResNet-152	384 × 288	Medium	0.695	0.859	0.766	0.771	0.929	0.839
ResNet-152	384 × 288	Large	0.817	0.937	0.882	0.874	0.981	0.931
HRNet-W32	256 × 192	Small	0.227	0.347	0.241	0.619	0.829	0.679
HRNet-W32	256 × 192	Medium	0.698	0.870	0.776	0.775	0.937	0.851
HRNet-W32	256 × 192	Large	0.812	0.942	0.887	0.870	0.982	0.933
HRNet-W32	384 × 288	Small	0.233	0.359	0.254	0.625	0.838	0.682
HRNet-W32	384 × 288	Medium	0.709	0.870	0.781	0.784	0.935	0.853
HRNet-W32	384 × 288	Large	0.831	0.944	0.894	0.884	0.984	0.938
HRNet-W48	256 × 192	Small	0.225	0.343	0.246	0.627	0.836	0.677
HRNet-W48	256 × 192	Medium	0.705	0.871	0.780	0.779	0.933	0.850
HRNet-W48	256 × 192	Large	0.819	0.944	0.890	0.874	0.982	0.934
HRNet-W48	384 × 288	Small	0.231	0.356	0.247	0.626	0.841	0.682
HRNet-W48	384 × 288	Medium	0.714	0.872	0.784	0.784	0.932	0.852
HRNet-W48	384 × 288	Large	0.837	0.946	0.896	0.889	0.984	0.939
DEKR-W32 ms	512 × 512	Small	0.158	0.264	0.164	0.509	0.748	0.539
DEKR-W32 ms	512 × 512	Medium	0.662	0.857	0.727	0.724	0.913	0.785
DEKR-W32 ms	512 × 512	Large	0.788	0.912	0.854	0.853	0.967	0.909
DEKR-W32	512 × 512	Small	0.143	0.263	0.143	0.429	0.675	0.458
DEKR-W32	512 × 512	Medium	0.632	0.851	0.696	0.685	0.894	0.745
DEKR-W32	512 × 512	Large	0.780	0.911	0.843	0.843	0.960	0.896
DEKR-W48 ms	640 × 640	Small	0.198	0.322	0.212	0.556	0.791	0.606
DEKR-W48 ms	640 × 640	Medium	0.687	0.863	0.755	0.747	0.920	0.811
DEKR-W48 ms	640 × 640	Large	0.790	0.905	0.847	0.864	0.971	0.913
DEKR-W48	640 × 640	Small	0.205	0.347	0.210	0.496	0.734	0.532
DEKR-W48	640 × 640	Medium	0.676	0.868	0.741	0.728	0.912	0.789
DEKR-W48	640 × 640	Large	0.786	0.909	0.844	0.856	0.969	0.905
DarkPose-W32	128 × 96	Small	0.216	0.331	0.245	0.598	0.824	0.675
DarkPose-W32	128 × 96	Medium	0.667	0.851	0.735	0.747	0.923	0.810
DarkPose-W32	128 × 96	Large	0.764	0.926	0.849	0.832	0.976	0.905
DarkPose-W32	256 × 192	Small	0.224	0.348	0.243	0.616	0.850	0.677
DarkPose-W32	256 × 192	Medium	0.705	0.867	0.775	0.782	0.938	0.850
DarkPose-W32	256 × 192	Large	0.829	0.939	0.891	0.884	0.981	0.936
DarkPose-W32	384 × 288	Small	0.222	0.342	0.240	0.619	0.829	0.665
DarkPose-W32	384 × 288	Medium	0.715	0.873	0.782	0.790	0.939	0.857
DarkPose-W32	384 × 288	Large	0.841	0.945	0.896	0.892	0.983	0.938
DarkPose-W48	128 × 96	Small	0.212	0.324	0.237	0.610	0.827	0.679
DarkPose-W48	128 × 96	Medium	0.678	0.856	0.754	0.756	0.927	0.827
DarkPose-W48	128 × 96	Large	0.777	0.925	0.855	0.842	0.973	0.912
DarkPose-W48	384 × 288	Small	0.238	0.361	0.253	0.630	0.846	0.691
DarkPose-W48	384 × 288	Medium	0.718	0.871	0.788	0.789	0.933	0.856
DarkPose-W48	384 × 288	Large	0.845	0.946	0.901	0.895	0.983	0.944

Table A.6: Results of all the tested methods for different resolutions on the validation set of the COCO Keypoint Detection Task 2017.

Method	Input size	Group	AP	AP^{50}	AP^{75}	AR	AR^{50}	AR^{75}
ResNet-50	256 × 192	Head	0.734	0.909	0.822	0.819	0.950	0.883
ResNet-50	256 × 192	Core body	0.776	0.909	0.835	0.836	0.949	0.883
ResNet-50	256 × 192	Arms	0.725	0.890	0.765	0.802	0.934	0.834
ResNet-50	256 × 192	Legs	0.716	0.855	0.740	0.785	0.910	0.805
ResNet-101	256 × 192	Head	0.739	0.914	0.824	0.828	0.951	0.891
ResNet-101	256 × 192	Core body	0.783	0.914	0.840	0.840	0.952	0.887
ResNet-101	256 × 192	Arms	0.740	0.895	0.783	0.817	0.940	0.852
ResNet-101	256 × 192	Legs	0.731	0.865	0.761	0.795	0.913	0.819
ResNet-152	256 × 192	Head	0.745	0.917	0.827	0.830	0.954	0.891
ResNet-152	256 × 192	Core body	0.788	0.913	0.845	0.844	0.951	0.891
ResNet-152	256 × 192	Arms	0.746	0.895	0.793	0.820	0.939	0.856
ResNet-152	256 × 192	Legs	0.736	0.864	0.767	0.800	0.914	0.825
ResNet-152	384 × 288	Head	0.791	0.924	0.854	0.859	0.957	0.905
ResNet-152	384 × 288	Core body	0.792	0.918	0.848	0.848	0.955	0.893
ResNet-152	384 × 288	Arms	0.774	0.906	0.810	0.838	0.946	0.867
ResNet-152	384 × 288	Legs	0.757	0.876	0.791	0.818	0.921	0.846
HRNet-W32	256 × 192	Head	0.785	0.926	0.865	0.857	0.959	0.912
HRNet-W32	256 × 192	Core body	0.797	0.922	0.855	0.852	0.957	0.899
HRNet-W32	256 × 192	Arms	0.768	0.909	0.810	0.837	0.949	0.871
HRNet-W32	256 × 192	Legs	0.748	0.877	0.782	0.812	0.924	0.839
HRNet-W32	384 × 288	Head	0.812	0.930	0.877	0.872	0.961	0.920
HRNet-W32	384 × 288	Core body	0.799	0.922	0.859	0.853	0.958	0.902
HRNet-W32	384 × 288	Arms	0.784	0.913	0.824	0.848	0.950	0.881
HRNet-W32	384 × 288	Legs	0.763	0.882	0.795	0.824	0.926	0.851
HRNet-W48	256 × 192	Head	0.796	0.931	0.873	0.864	0.961	0.919
HRNet-W48	256 × 192	Core body	0.796	0.923	0.854	0.852	0.958	0.898
HRNet-W48	256 × 192	Arms	0.780	0.912	0.822	0.844	0.949	0.876
HRNet-W48	256 × 192	Legs	0.758	0.878	0.786	0.819	0.923	0.841
HRNet-W48	384 × 288	Head	0.823	0.933	0.886	0.879	0.960	0.924
HRNet-W48	384 × 288	Core body	0.798	0.923	0.857	0.852	0.955	0.899
HRNet-W48	384 × 288	Arms	0.790	0.914	0.830	0.850	0.948	0.882
HRNet-W48	384 × 288	Legs	0.772	0.886	0.797	0.829	0.929	0.850
DEKR-W32 ms	512 × 512	Head	0.786	0.907	0.841	0.840	0.939	0.881
DEKR-W32 ms	512 × 512	Core body	0.767	0.904	0.824	0.817	0.938	0.862
DEKR-W32 ms	512 × 512	Arms	0.720	0.876	0.757	0.787	0.915	0.817
DEKR-W32 ms	512 × 512	Legs	0.719	0.864	0.747	0.787	0.911	0.810
DEKR-W32	512 × 512	Head	0.745	0.892	0.797	0.807	0.921	0.848
DEKR-W32	512 × 512	Core body	0.756	0.896	0.811	0.803	0.925	0.847
DEKR-W32	512 × 512	Arms	0.688	0.861	0.720	0.755	0.897	0.781
DEKR-W32	512 × 512	Legs	0.698	0.849	0.725	0.756	0.887	0.776
DEKR-W48 ms	640 × 640	Head	0.806	0.915	0.856	0.858	0.948	0.899
DEKR-W48 ms	640 × 640	Core body	0.772	0.905	0.833	0.825	0.943	0.873
DEKR-W48 ms	640 × 640	Arms	0.737	0.883	0.773	0.805	0.926	0.833
DEKR-W48 ms	640 × 640	Legs	0.739	0.875	0.766	0.805	0.921	0.829
DEKR-W48	640 × 640	Head	0.790	0.914	0.841	0.844	0.941	0.883
DEKR-W48	640 × 640	Core body	0.768	0.904	0.826	0.816	0.934	0.863
DEKR-W48	640 × 640	Arms	0.723	0.879	0.759	0.789	0.917	0.819
DEKR-W48	640 × 640	Legs	0.723	0.867	0.751	0.780	0.902	0.801
DarkPose-W32	128 × 96	Head	0.741	0.908	0.814	0.828	0.948	0.885
DarkPose-W32	128 × 96	Core body	0.776	0.909	0.835	0.837	0.949	0.887
DarkPose-W32	128 × 96	Arms	0.713	0.886	0.750	0.798	0.932	0.830
DarkPose-W32	128 × 96	Legs	0.714	0.850	0.746	0.783	0.904	0.810
DarkPose-W32	256 × 192	Head	0.817	0.929	0.875	0.875	0.961	0.917
DarkPose-W32	256 × 192	Core body	0.798	0.920	0.858	0.853	0.957	0.901
DarkPose-W32	256 × 192	Arms	0.778	0.909	0.821	0.842	0.945	0.875
DarkPose-W32	256 × 192	Legs	0.758	0.876	0.788	0.818	0.923	0.843
DarkPose-W32	384 × 288	Head	0.831	0.929	0.885	0.883	0.960	0.924
DarkPose-W32	384 × 288	Core body	0.805	0.925	0.861	0.857	0.958	0.902
DarkPose-W32	384 × 288	Arms	0.790	0.913	0.828	0.851	0.950	0.881
DarkPose-W32	384 × 288	Legs	0.773	0.890	0.799	0.831	0.932	0.852
DarkPose-W48	128 × 96	Head	0.758	0.913	0.833	0.841	0.950	0.896
DarkPose-W48	128 × 96	Core body	0.782	0.913	0.841	0.842	0.954	0.891
DarkPose-W48	128 × 96	Arms	0.731	0.892	0.772	0.810	0.937	0.843
DarkPose-W48	128 × 96	Legs	0.732	0.862	0.763	0.796	0.913	0.821
DarkPose-W48	384 × 288	Head	0.835	0.930	0.888	0.884	0.960	0.926
DarkPose-W48	384 × 288	Core body	0.805	0.924	0.862	0.857	0.957	0.904
DarkPose-W48	384 × 288	Arms	0.794	0.914	0.833	0.853	0.948	0.885
DarkPose-W48	384 × 288	Legs	0.775	0.885	0.804	0.832	0.928	0.856

Table A.7: Results of all the tested methods for different types of grouped keypoints on the validation set of the COCO Keypoint Detection Task 2017.

Method	Input size	Group	AP	AP^{50}	AP^{75}	AR	AR^{50}	AR^{75}
ResNet-50	256 × 192	Head	0.746	0.917	0.836	0.830	0.954	0.895
ResNet-50	256 × 192	Core body	0.825	0.926	0.882	0.881	0.962	0.925
ResNet-50	256 × 192	Arms	0.786	0.912	0.830	0.856	0.952	0.888
ResNet-50	256 × 192	Legs	0.775	0.881	0.805	0.842	0.934	0.863
ResNet-101	256 × 192	Head	0.750	0.918	0.837	0.837	0.956	0.901
ResNet-101	256 × 192	Core body	0.828	0.927	0.884	0.883	0.963	0.926
ResNet-101	256 × 192	Arms	0.798	0.915	0.847	0.866	0.955	0.902
ResNet-101	256 × 192	Legs	0.788	0.893	0.819	0.849	0.939	0.872
ResNet-152	256 × 192	Head	0.757	0.923	0.842	0.841	0.958	0.902
ResNet-152	256 × 192	Core body	0.833	0.927	0.887	0.886	0.961	0.928
ResNet-152	256 × 192	Arms	0.804	0.914	0.850	0.870	0.954	0.903
ResNet-152	256 × 192	Legs	0.791	0.890	0.821	0.854	0.938	0.876
ResNet-152	384 × 288	Head	0.803	0.930	0.868	0.869	0.962	0.916
ResNet-152	384 × 288	Core body	0.838	0.930	0.890	0.890	0.965	0.929
ResNet-152	384 × 288	Arms	0.829	0.921	0.871	0.886	0.958	0.916
ResNet-152	384 × 288	Legs	0.814	0.901	0.846	0.872	0.945	0.897
HRNet-W32	256 × 192	Head	0.798	0.934	0.877	0.868	0.964	0.922
HRNet-W32	256 × 192	Core body	0.842	0.934	0.898	0.893	0.966	0.935
HRNet-W32	256 × 192	Arms	0.826	0.923	0.869	0.885	0.959	0.917
HRNet-W32	256 × 192	Legs	0.807	0.902	0.843	0.866	0.945	0.893
HRNet-W32	384 × 288	Head	0.826	0.936	0.893	0.884	0.965	0.931
HRNet-W32	384 × 288	Core body	0.844	0.934	0.897	0.894	0.967	0.935
HRNet-W32	384 × 288	Arms	0.838	0.928	0.879	0.893	0.961	0.923
HRNet-W32	384 × 288	Legs	0.820	0.906	0.850	0.876	0.949	0.901
HRNet-W48	256 × 192	Head	0.807	0.935	0.887	0.873	0.965	0.929
HRNet-W48	256 × 192	Core body	0.840	0.935	0.896	0.891	0.966	0.932
HRNet-W48	256 × 192	Arms	0.837	0.930	0.879	0.891	0.962	0.921
HRNet-W48	256 × 192	Legs	0.814	0.902	0.846	0.871	0.944	0.895
HRNet-W48	384 × 288	Head	0.836	0.938	0.898	0.889	0.965	0.934
HRNet-W48	384 × 288	Core body	0.844	0.935	0.898	0.893	0.965	0.933
HRNet-W48	384 × 288	Arms	0.842	0.929	0.880	0.894	0.960	0.921
HRNet-W48	384 × 288	Legs	0.825	0.910	0.852	0.878	0.950	0.897
DEKR-W32 ms	512 × 512	Head	0.799	0.915	0.856	0.851	0.944	0.893
DEKR-W32 ms	512 × 512	Core body	0.815	0.920	0.869	0.863	0.953	0.904
DEKR-W32 ms	512 × 512	Arms	0.783	0.899	0.819	0.841	0.936	0.869
DEKR-W32 ms	512 × 512	Legs	0.773	0.881	0.803	0.835	0.927	0.858
DEKR-W32	512 × 512	Head	0.756	0.897	0.809	0.819	0.928	0.858
DEKR-W32	512 × 512	Core body	0.801	0.911	0.855	0.846	0.940	0.888
DEKR-W32	512 × 512	Arms	0.751	0.885	0.787	0.810	0.918	0.837
DEKR-W32	512 × 512	Legs	0.751	0.867	0.780	0.804	0.903	0.826
DEKR-W48 ms	640 × 640	Head	0.818	0.923	0.867	0.870	0.954	0.909
DEKR-W48 ms	640 × 640	Core body	0.820	0.919	0.874	0.871	0.956	0.912
DEKR-W48 ms	640 × 640	Arms	0.798	0.903	0.834	0.858	0.944	0.886
DEKR-W48 ms	640 × 640	Legs	0.792	0.894	0.820	0.852	0.938	0.874
DEKR-W48	640 × 640	Head	0.801	0.918	0.852	0.856	0.947	0.893
DEKR-W48	640 × 640	Core body	0.814	0.918	0.865	0.860	0.949	0.900
DEKR-W48	640 × 640	Arms	0.783	0.901	0.819	0.841	0.936	0.869
DEKR-W48	640 × 640	Legs	0.774	0.886	0.800	0.826	0.921	0.846
DarkPose-W32	128 × 96	Head	0.752	0.915	0.824	0.838	0.953	0.893
DarkPose-W32	128 × 96	Core body	0.822	0.925	0.880	0.879	0.961	0.924
DarkPose-W32	128 × 96	Arms	0.776	0.906	0.822	0.852	0.949	0.885
DarkPose-W32	128 × 96	Legs	0.770	0.880	0.803	0.836	0.930	0.861
DarkPose-W32	256 × 192	Head	0.829	0.933	0.886	0.885	0.964	0.926
DarkPose-W32	256 × 192	Core body	0.842	0.933	0.895	0.893	0.966	0.933
DarkPose-W32	256 × 192	Arms	0.832	0.923	0.872	0.887	0.957	0.917
DarkPose-W32	256 × 192	Legs	0.813	0.901	0.840	0.869	0.945	0.890
DarkPose-W32	384 × 288	Head	0.844	0.936	0.897	0.892	0.964	0.933
DarkPose-W32	384 × 288	Core body	0.849	0.936	0.899	0.896	0.968	0.934
DarkPose-W32	384 × 288	Arms	0.843	0.929	0.880	0.894	0.962	0.920
DarkPose-W32	384 × 288	Legs	0.826	0.911	0.850	0.879	0.951	0.898
DarkPose-W48	128 × 96	Head	0.770	0.919	0.843	0.851	0.956	0.905
DarkPose-W48	128 × 96	Core body	0.827	0.927	0.882	0.883	0.965	0.926
DarkPose-W48	128 × 96	Arms	0.788	0.909	0.831	0.859	0.951	0.890
DarkPose-W48	128 × 96	Legs	0.781	0.888	0.808	0.845	0.937	0.865
DarkPose-W48	384 × 288	Head	0.848	0.938	0.900	0.895	0.965	0.935
DarkPose-W48	384 × 288	Core body	0.848	0.937	0.900	0.895	0.966	0.934
DarkPose-W48	384 × 288	Arms	0.846	0.930	0.882	0.897	0.961	0.924
DarkPose-W48	384 × 288	Legs	0.827	0.908	0.854	0.880	0.949	0.901

Table A.8: Results of all the tested methods for different types of visible keypoints groups on the validation set of the COCO Keypoint Detection Task 2017.

Method	Input size	Group	AP	AP^{50}	AP^{75}	AR	AR^{50}	AR^{75}
ResNet-50	256 × 192	Head	0.419	0.627	0.450	0.636	0.831	0.671
ResNet-50	256 × 192	Core body	0.586	0.797	0.638	0.717	0.890	0.766
ResNet-50	256 × 192	Arms	0.421	0.648	0.423	0.592	0.794	0.606
ResNet-50	256 × 192	Legs	0.430	0.621	0.446	0.584	0.774	0.604
ResNet-101	256 × 192	Head	0.441	0.645	0.461	0.666	0.845	0.698
ResNet-101	256 × 192	Core body	0.598	0.807	0.651	0.724	0.901	0.773
ResNet-101	256 × 192	Arms	0.440	0.671	0.452	0.617	0.813	0.642
ResNet-101	256 × 192	Legs	0.460	0.644	0.490	0.605	0.782	0.638
ResNet-152	256 × 192	Head	0.437	0.632	0.473	0.654	0.829	0.700
ResNet-152	256 × 192	Core body	0.607	0.816	0.656	0.731	0.908	0.776
ResNet-152	256 × 192	Arms	0.450	0.681	0.464	0.615	0.814	0.642
ResNet-152	256 × 192	Legs	0.463	0.652	0.487	0.604	0.790	0.628
ResNet-152	384 × 288	Head	0.489	0.683	0.520	0.686	0.851	0.719
ResNet-152	384 × 288	Core body	0.613	0.822	0.663	0.732	0.906	0.774
ResNet-152	384 × 288	Arms	0.496	0.710	0.513	0.654	0.839	0.672
ResNet-152	384 × 288	Legs	0.482	0.678	0.502	0.622	0.801	0.647
HRNet-W32	256 × 192	Head	0.474	0.675	0.502	0.678	0.845	0.709
HRNet-W32	256 × 192	Core body	0.614	0.824	0.670	0.740	0.909	0.792
HRNet-W32	256 × 192	Arms	0.485	0.712	0.506	0.648	0.835	0.678
HRNet-W32	256 × 192	Legs	0.466	0.663	0.480	0.616	0.803	0.635
HRNet-W32	384 × 288	Head	0.504	0.691	0.547	0.687	0.860	0.729
HRNet-W32	384 × 288	Core body	0.620	0.834	0.677	0.742	0.913	0.793
HRNet-W32	384 × 288	Arms	0.509	0.726	0.541	0.663	0.842	0.699
HRNet-W32	384 × 288	Legs	0.490	0.693	0.508	0.636	0.821	0.660
HRNet-W48	256 × 192	Head	0.502	0.685	0.536	0.703	0.864	0.750
HRNet-W48	256 × 192	Core body	0.618	0.831	0.671	0.742	0.913	0.789
HRNet-W48	256 × 192	Arms	0.495	0.716	0.521	0.650	0.834	0.681
HRNet-W48	256 × 192	Legs	0.488	0.685	0.510	0.631	0.809	0.660
HRNet-W48	384 × 288	Head	0.524	0.711	0.571	0.716	0.874	0.764
HRNet-W48	384 × 288	Core body	0.621	0.830	0.672	0.741	0.911	0.787
HRNet-W48	384 × 288	Arms	0.522	0.741	0.541	0.667	0.847	0.695
HRNet-W48	384 × 288	Legs	0.508	0.709	0.525	0.648	0.829	0.671
DEKR-W32 ms	512 × 512	Head	0.456	0.633	0.467	0.634	0.808	0.645
DEKR-W32 ms	512 × 512	Core body	0.591	0.807	0.642	0.694	0.879	0.740
DEKR-W32 ms	512 × 512	Arms	0.439	0.668	0.441	0.582	0.781	0.600
DEKR-W32 ms	512 × 512	Legs	0.447	0.663	0.462	0.601	0.795	0.617
DEKR-W32	512 × 512	Head	0.421	0.609	0.427	0.602	0.783	0.622
DEKR-W32	512 × 512	Core body	0.579	0.792	0.625	0.673	0.858	0.711
DEKR-W32	512 × 512	Arms	0.411	0.636	0.413	0.553	0.756	0.568
DEKR-W32	512 × 512	Legs	0.437	0.652	0.442	0.563	0.765	0.574
DEKR-W48 ms	640 × 640	Head	0.468	0.647	0.496	0.652	0.824	0.686
DEKR-W48 ms	640 × 640	Core body	0.598	0.812	0.647	0.703	0.887	0.746
DEKR-W48 ms	640 × 640	Arms	0.469	0.685	0.488	0.610	0.799	0.635
DEKR-W48 ms	640 × 640	Legs	0.477	0.690	0.490	0.630	0.815	0.649
DEKR-W48	640 × 640	Head	0.442	0.620	0.449	0.634	0.802	0.655
DEKR-W48	640 × 640	Core body	0.597	0.810	0.649	0.693	0.874	0.741
DEKR-W48	640 × 640	Arms	0.449	0.670	0.458	0.587	0.786	0.606
DEKR-W48	640 × 640	Legs	0.471	0.681	0.485	0.599	0.788	0.616
DarkPose-W32	128 × 96	Head	0.422	0.634	0.455	0.652	0.847	0.688
DarkPose-W32	128 × 96	Core body	0.587	0.800	0.635	0.722	0.900	0.769
DarkPose-W32	128 × 96	Arms	0.403	0.631	0.406	0.592	0.793	0.610
DarkPose-W32	128 × 96	Legs	0.440	0.634	0.454	0.594	0.781	0.614
DarkPose-W32	256 × 192	Head	0.500	0.688	0.539	0.693	0.864	0.731
DarkPose-W32	256 × 192	Core body	0.623	0.826	0.674	0.746	0.913	0.792
DarkPose-W32	256 × 192	Arms	0.503	0.713	0.534	0.655	0.831	0.692
DarkPose-W32	256 × 192	Legs	0.488	0.676	0.507	0.632	0.809	0.657
DarkPose-W32	384 × 288	Head	0.522	0.696	0.565	0.708	0.870	0.750
DarkPose-W32	384 × 288	Core body	0.628	0.833	0.673	0.748	0.913	0.788
DarkPose-W32	384 × 288	Arms	0.518	0.734	0.535	0.672	0.851	0.699
DarkPose-W32	384 × 288	Legs	0.512	0.708	0.529	0.655	0.834	0.679
DarkPose-W48	128 × 96	Head	0.429	0.643	0.445	0.654	0.849	0.696
DarkPose-W48	128 × 96	Core body	0.596	0.809	0.638	0.729	0.909	0.770
DarkPose-W48	128 × 96	Arms	0.430	0.664	0.430	0.609	0.814	0.626
DarkPose-W48	128 × 96	Legs	0.476	0.671	0.501	0.620	0.801	0.643
DarkPose-W48	384 × 288	Head	0.526	0.700	0.575	0.711	0.870	0.762
DarkPose-W48	384 × 288	Core body	0.632	0.838	0.686	0.751	0.915	0.798
DarkPose-W48	384 × 288	Arms	0.535	0.748	0.561	0.676	0.851	0.713
DarkPose-W48	384 × 288	Legs	0.521	0.718	0.535	0.657	0.837	0.675

Table A.9: Results of all the tested methods for different types of occluded keypoints groups on the validation set of the COCO Keypoint Detection Task 2017.